



University of
Zurich^{UZH}

Zurich Open Repository and
Archive

University of Zurich
University Library
Strickhofstrasse 39
CH-8057 Zurich
www.zora.uzh.ch

Year: 2020

CP2K: An electronic structure and molecular dynamics software package - Quickstep: Efficient and accurate electronic structure calculations

Kühne, Thomas D ; Iannuzzi, Marcella ; Del Ben, Mauro ; Rybkin, Vladimir V ; Seewald, Patrick ; Stein, Frederick ; Laino, Teodoro ; Khaliullin, Rustam Z ; Schütt, Ole ; Schiffmann, Florian ; Golze, Dorothea ; Wilhelm, Jan ; Chulkov, Sergey ; Bani-Hashemian, Mohammad Hossein ; Weber, Valéry ; Borštnik, Urban ; Taillefumier, Mathieu ; Jakobovits, Alice Shoshana ; Lazzaro, Alfio ; Pabst, Hans ; Müller, Tiziano ; Schade, Robert ; Guidon, Manuel ; Andermatt, Samuel ; et al ; Hehn, Anna ; Bussy, Augustin ; Belleflamme, Fabian ; VandeVondele, Joost ; Hutter, Jürg

Abstract: CP2K is an open source electronic structure and molecular dynamics software package to perform atomistic simulations of solid-state, liquid, molecular, and biological systems. It is especially aimed at massively parallel and linear-scaling electronic structure methods and state-of-the-art ab initio molecular dynamics simulations. Excellent performance for electronic structure calculations is achieved using novel algorithms implemented for modern high-performance computing systems. This review revisits the main capabilities of CP2K to perform efficient and accurate electronic structure simulations. The emphasis is put on density functional theory and multiple post-Hartree-Fock methods using the Gaussian and plane wave approach and its augmented all-electron extension.

DOI: <https://doi.org/10.1063/5.0007045>

Posted at the Zurich Open Repository and Archive, University of Zurich

ZORA URL: <https://doi.org/10.5167/uzh-192979>

Journal Article

Published Version



The following work is licensed under a Creative Commons: Attribution 4.0 International (CC BY 4.0) License.

Originally published at:

Kühne, Thomas D ; Iannuzzi, Marcella ; Del Ben, Mauro ; Rybkin, Vladimir V ; Seewald, Patrick ; Stein, Frederick ; Laino, Teodoro ; Khaliullin, Rustam Z ; Schütt, Ole ; Schiffmann, Florian ; Golze, Dorothea ; Wilhelm, Jan ; Chulkov, Sergey ; Bani-Hashemian, Mohammad Hossein ; Weber, Valéry ; Borštnik, Urban ; Taillefumier, Mathieu ; Jakobovits, Alice Shoshana ; Lazzaro, Alfio ; Pabst, Hans ; Müller, Tiziano ; Schade, Robert ; Guidon, Manuel ; Andermatt, Samuel ; et al ; Hehn, Anna ; Bussy, Augustin ; Belleflamme, Fabian ; VandeVondele, Joost ; Hutter, Jürg (2020). CP2K: An electronic structure and molecular dynamics software package - Quickstep: Efficient and accurate electronic structure calculations. *Journal of Chemical Physics*, 152(19):194103.

DOI: <https://doi.org/10.1063/5.0007045>

CP2K: An electronic structure and molecular dynamics software package - Quickstep: Efficient and accurate electronic structure calculations ^{EP}

Cite as: J. Chem. Phys. **152**, 194103 (2020); <https://doi.org/10.1063/5.0007045>

Submitted: 10 March 2020 . Accepted: 22 April 2020 . Published Online: 19 May 2020

 Thomas D. Kühne,  Marcella Iannuzzi, Mauro Del Ben,  Vladimir V. Rybkin, Patrick Seewald, Frederick Stein,  Teodoro Laino,  Rustam Z. Khaliullin, Ole Schütt,  Florian Schiffmann,  Dorothea Golze, Jan Wilhelm, Sergey Chulkov,  Mohammad Hossein Bani-Hashemian, Valéry Weber, Urban Borštnik, Mathieu Taillefumier,  Alice Shoshana Jakobovits, Alfio Lazzaro, Hans Pabst, Tiziano Müller,  Robert Schade, Manuel Guidon, Samuel Andermatt, Nico Holmberg,  Gregory K. Schenter, Anna Hehn, Augustin Bussy, Fabian Belleflamme,  Gloria Tabacchi, Andreas Glöß,  Michael Lass,  Iain Bethune,  Christopher J. Mundy,  Christian Plessl,  Matt Watkins,  Joost VandeVondele,  Matthias Krack, and Jürg Hutter

COLLECTIONS

Paper published as part of the special topic on [Electronic Structure SoftwareESS2020](#)

 This paper was selected as an Editor's Pick



View Online



Export Citation



CrossMark

ARTICLES YOU MAY BE INTERESTED IN

[A consistent and accurate ab initio parametrization of density functional dispersion correction \(DFT-D\) for the 94 elements H-Pu](#)

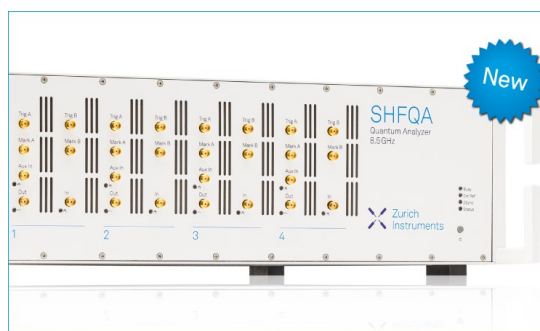
The Journal of Chemical Physics **132**, 154104 (2010); <https://doi.org/10.1063/1.3382344>

[DFTB+, a software package for efficient approximate density functional theory based atomistic simulations](#)

The Journal of Chemical Physics **152**, 124101 (2020); <https://doi.org/10.1063/1.5143190>

[Quantum ESPRESSO toward the exascale](#)

The Journal of Chemical Physics **152**, 154105 (2020); <https://doi.org/10.1063/5.0005082>



Your Qubits. Measured.

Meet the next generation of quantum analyzers

- Readout for up to 64 qubits
- Operation at up to 8.5 GHz, mixer-calibration-free
- Signal optimization with minimal latency

Find out more



CP2K: An electronic structure and molecular dynamics software package - Quickstep: Efficient and accurate electronic structure calculations

Cite as: J. Chem. Phys. 152, 194103 (2020); doi: 10.1063/5.0007045

Submitted: 10 March 2020 • Accepted: 22 April 2020 •

Published Online: 19 May 2020



Thomas D. Kühne,^{1,a)}  Marcella Iannuzzi,²  Mauro Del Ben,³ Vladimir V. Rybkin,²  Patrick Seewald,² Frederick Stein,² Teodoro Laino,⁴  Rustam Z. Khaliullin,⁵  Ole Schütt,⁶ Florian Schiffmann,⁷  Dorothea Golze,⁸  Jan Wilhelm,⁹ Sergey Chulkov,¹⁰ Mohammad Hossein Bani-Hashemian,¹¹  Valéry Weber,⁴ Urban Borštnik,¹² Mathieu Taillefumier,¹³ Alice Shoshana Jakobovits,¹³  Alfio Lazzaro,¹⁴ Hans Pabst,¹⁵ Tiziano Müller,² Robert Schade,¹⁶  Manuel Cuidon,² Samuel Andermatt,¹¹ Nico Holmberg,¹⁷ Gregory K. Schenter,¹⁸  Anna Hehn,² Augustin Bussy,² Fabian Belleflamme,² Gloria Tabacchi,¹⁹  Andreas Glöß,²⁰ Michael Lass,¹⁶  Iain Bethune,²¹  Christopher J. Mundy,¹⁸  Christian Plessl,¹⁶  Matt Watkins,¹⁰  Joost VandeVondele,¹³  Matthias Krack,^{22,b)}  and Jürg Hutter^{2,c)}

AFFILIATIONS

¹ Dynamics of Condensed Matter and Center for Sustainable Systems Design, Chair of Theoretical Chemistry, Paderborn University, Warburger Str. 100, D-33098 Paderborn, Germany

² Department of Chemistry, University of Zurich, Winterthurerstrasse 190, CH-8057 Zurich, Switzerland

³ Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA

⁴ IBM Research Europe, CH-8803 Rüschlikon, Switzerland

⁵ Department of Chemistry, McGill University, CH-801 Sherbrooke St. West, Montreal, Quebec H3A 0B8, Canada

⁶ Department of Materials, ETH Zurich, CH-8092 Zurich, Switzerland

⁷ Centre of Policy Studies, Victoria University, Melbourne, Australia

⁸ Department of Applied Physics, Aalto University, Otakaari 1, FI-02150 Espoo, Finland

⁹ Institute of Theoretical Physics, University of Regensburg, Universitätsstraße 31, D-93053 Regensburg, Germany

¹⁰ School of Mathematics and Physics, University of Lincoln, Brayford Pool, Lincoln, United Kingdom

¹¹ Integrated Systems Laboratory, ETH Zurich, CH-8092 Zurich, Switzerland

¹² Scientific IT Services, ETH Zurich, Zurich, Switzerland

¹³ Swiss National Supercomputing Centre (CSCS), ETH Zurich, Zurich, Switzerland

¹⁴ HPE Switzerland GmbH, Basel, Switzerland

¹⁵ Intel Extreme Computing, Software and Systems, Zurich, Switzerland

¹⁶ Department of Computer Science and Paderborn Center for Parallel Computing, Paderborn University, Warburger Str. 100, D-33098 Paderborn, Germany

¹⁷ Department of Chemistry and Materials Science, Aalto University, P.O. Box 16100, 00076 Aalto, Finland

¹⁸ Physical Science Division, Pacific Northwest National Laboratory, P.O. Box 999, Richland, Washington 99352, USA

¹⁹ Department of Science and High Technology, University of Insubria and INSTM, via Valleggio 9, I-22100 Como, Italy

²⁰ BASF SE, Carl-Bosch-Straße 38, D-67056 Ludwigshafen am Rhein, Germany

²¹ Hartree Centre, Science and Technology Facilities Council, Sci-Tech Daresbury, Warrington WA4 4AD, United Kingdom

²² Laboratory for Scientific Computing and Modelling, Paul Scherrer Institute, CH-5232 Villigen PSI, Switzerland

Note: This article is part of the JCP Special Topic on Electronic Structure Software.

^{a)} Author to whom correspondence should be addressed: tdkuehne@mail.uni-paderborn.de

^{b)} matthias.krack@psi.ch

^{c)} hutter@chem.uzh.ch

ABSTRACT

CP2K is an open source electronic structure and molecular dynamics software package to perform atomistic simulations of solid-state, liquid, molecular, and biological systems. It is especially aimed at massively parallel and linear-scaling electronic structure methods and state-of-the-art *ab initio* molecular dynamics simulations. Excellent performance for electronic structure calculations is achieved using novel algorithms implemented for modern high-performance computing systems. This review revisits the main capabilities of CP2K to perform efficient and accurate electronic structure simulations. The emphasis is put on density functional theory and multiple post-Hartree-Fock methods using the Gaussian and plane wave approach and its augmented all-electron extension.

© 2020 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>). <https://doi.org/10.1063/5.0007045>

I. INTRODUCTION

The geometric increase in the performance of computers over the last few decades, together with advances in theoretical methods and applied mathematics, has established computational science as an indispensable technique in chemistry, physics, and life and materials sciences. In fact, computer simulations have been very successful in explaining a large variety of new scientific phenomena, interpret experimental measurements, predict materials properties, and even rationally design new systems. Therefore, conducting experiments *in silico* permits to investigate systems atomistically that otherwise would be too difficult, expensive, or simply impossible to perform. However, the by far most rewarding outcome of such simulations is the invaluable insights they provide into the atomistic behavior and the dynamics. Therefore, electronic structure theory based *ab initio* molecular dynamics (AIMD) can be thought of as a computational microscope.^{1–3}

The open source electronic structure and molecular dynamics (MD) software package CP2K aims to provide a broad range of computational methods and simulation approaches suitable for extended condensed-phase systems. The latter is made possible by combining efficient algorithms with excellent parallel scalability to exploit modern high-performance computing architectures. However, along with conducting efficient large-scale AIMD simulations, CP2K provides a much broader range of capabilities, which includes the possibility of choosing the most adequate approach for a given problem and the flexibility of combining computational methods.

The remainder of this paper is organized as follows: The Gaussian and plane wave (GPW) approach to density functional theory (DFT) is reviewed in Sec. II and before Hartree-Fock and beyond Hartree-Fock methods are covered in Secs. III and IV, respectively. Thereafter, constrained DFT (CDFT), density functional perturbation theory (DFPT), and time-dependent DFT (TD-DFT) are described in Secs. V, VI, and VII, respectively. Sections VIII and XIII are devoted to low-scaling eigenvalue solver based on sparse matrix linear algebra using the distributed block compressed sparse row (DBCSPR) library. Conventional orthogonal localized orbitals, non-orthogonal localized orbitals (NLMOs), absolutely localized molecular orbitals (ALMOs), and compact localized molecular orbitals (CLMOs) are discussed in Sec. IX to facilitate linear-scaling AIMD, whose key concepts are detailed in Sec. X. Energy decomposition and spectroscopic analysis methods are presented in Sec. XI, followed by various embedding techniques, which are summarized in

Sec. XII. Interfaces to other programs and technical aspects of CP2K are specified in Secs. XIV and XV, respectively.

II. GAUSSIAN AND PLANE WAVE METHOD

The electronic structure module QUICKSTEP^{4,5} in CP2K can handle a wide spectrum of methods and approaches. Semi-empirical (SE) and tight-binding (TB) methods, orbital-free and Kohn-Sham DFT (KS-DFT) and wavefunction-based correlation methods [e.g., second-order Møller-Plesset perturbation theory (MP2), direct-random phase approximation (dRPA), and GW] all make use of the same infrastructure of integral routines and optimization algorithms. In this section, we give a brief overview of the methodology that sets CP2K apart from most other electronic structure programs, namely, its use of a plane wave (PW) auxiliary basis set within a Gaussian orbital scheme. As many other programs, CP2K uses contracted Gaussian basis sets $g(\mathbf{r})$ to expand orbital functions

$$\varphi(\mathbf{r}) = \sum_u d_u g_u(\mathbf{r}), \quad (1)$$

where the contraction coefficients d_u are fixed and the primitive Gaussians

$$g(\mathbf{r}) = r^l \exp[-\alpha(\mathbf{r} - \mathbf{A})^2] Y_{lm}(\mathbf{r} - \mathbf{A}) \quad (2)$$

are centered at atomic positions. These functions are defined by the exponent α , the spherical harmonics Y_{lm} with angular momentum (l, m), and the coordinates of its center \mathbf{A} . The unique properties of Gaussians, e.g., analytic integration or the product theorem, are exploited in many programs. In CP2K, we make use of an additional property of Gaussians, namely, that their Fourier transform is again a Gaussian function, i.e.,

$$\int \exp[-\alpha \mathbf{r}^2] \exp[-i\mathbf{G} \cdot \mathbf{r}] d\mathbf{r} = \exp\left[-\frac{\mathbf{G}^2}{4\alpha}\right]. \quad (3)$$

This property is directly connected with the fact that integration of Gaussian functions on equidistant grids shows exponential convergence with grid spacing. In order to take advantage of this property, we define, within a computational box or periodic unit cell, a set of equidistant grid points

$$\mathbf{R} = \mathbf{h} \mathbf{N} \mathbf{q}. \quad (4)$$

The three vectors \mathbf{a}_1 , \mathbf{a}_2 , and \mathbf{a}_3 define a computational box with a 3×3 matrix $\mathbf{h} = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3]$ and a volume $\Omega = \det \mathbf{h}$. Furthermore,

N is a diagonal matrix with entries $1/N_s$, where N_s is the number of grid points along vector $s = 1, 2, 3$, whereas \mathbf{q} is a vector of integers ranging from 0 to $N_s - 1$. Reciprocal lattice vectors \mathbf{b}_i , defined by $\mathbf{b}_i \cdot \mathbf{a}_j = 2\pi\delta_{ij}$, can also be arranged into a 3×3 matrix $[\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3] = 2\pi(\mathbf{h}^t)^{-1}$, which allows us to define reciprocal space vectors

$$\mathbf{G} = 2\pi(\mathbf{h}^t)^{-1}\mathbf{g}, \quad (5)$$

where \mathbf{g} is a vector of integer values. Any function with periodicity given by the lattice vectors and defined on the real-space points \mathbf{R} can be transformed into a reciprocal space representation by the Fourier transform

$$f(\mathbf{G}) = \sum_{\mathbf{R}} f(\mathbf{R}) \exp[i\mathbf{G} \cdot \mathbf{R}]. \quad (6)$$

The accuracy of this expansion is given by the grid spacings or the PW cutoff defining the largest vector \mathbf{G} included in the sum.

In the GPW method,⁶ the equidistant grid, or equivalently the PW expansion within the computational box, is used for an alternative representation of the electron density. In the KS method, the electron density is defined by

$$n(\mathbf{r}) = \sum_{\mu\nu} P_{\mu\nu} \varphi_{\mu}(\mathbf{r}) \varphi_{\nu}(\mathbf{r}), \quad (7)$$

where the density matrix \mathbf{P} , with elements $P_{\mu\nu} = \sum_i f_i c_{\mu i} c_{\nu i}$, is calculated from the orbital occupations f_i and the orbital expansion coefficients $c_{\mu i}$ of the common linear combination of atomic orbitals $\Phi_i(\mathbf{r}) = \sum_{\mu} c_{\mu i} \varphi_{\mu}(\mathbf{r})$. Therein, $\Phi_i(\mathbf{r})$ are the so-called molecular orbitals (MOs) and $\varphi_{\mu}(\mathbf{r})$ are the atomic orbitals (AOs). In the PW expansion, however, the density is given by

$$n(\mathbf{r}) = \sum_{\mathbf{G}} n(\mathbf{G}) \exp[i\mathbf{G} \cdot \mathbf{r}]. \quad (8)$$

The definitions given above allow us to calculate the expansion coefficients $n(\mathbf{G})$ from the density matrix $P_{\mu\nu}$ and the basis functions $\varphi_{\mu}(\mathbf{r})$. The dual representation of the density is used in the definition of the GPW-based KS energy expression (see Sec. II A) to facilitate efficient and accurate algorithms for the electrostatic as well as exchange and correlation energies. The efficient mapping $P_{\mu\nu} \rightarrow n(\mathbf{G})$ is achieved by using multigrid methods, optimal screening in real-space, and the separability of Cartesian Gaussian functions in orthogonal coordinates. Details of these algorithms that result in a linear scaling algorithm with a small prefactor for the mapping is described elsewhere.⁴⁻⁶

A. Kohn-Sham energy, forces, and stress tensor

The electronic KS energy functional⁷ for a molecular or crystalline system within the supercell or Γ -point approximation in the GPW framework⁴⁻⁶ is defined as

$$E[\mathbf{P}] = E^{\text{kin}}[\mathbf{P}] + E^{\text{ext}}[\mathbf{P}] + E^{\text{ES}}[n_{\mathbf{G}}] + E^{\text{XC}}[n_{\mathbf{G}}], \quad (9)$$

where E^{kin} is the kinetic energy, E^{ext} is the electronic interaction with the ionic cores (see Sec. II B), E^{ES} is the total electrostatic (Coulomb) energy, and E^{XC} is the exchange-correlation (XC) energy. An extension of Eq. (9) to include a k-point sampling within the first Brillouin zone is also available in CP2K. This implementation follows the

methods from Pisani,⁸ Kudin,⁹ and co-workers. The electrostatic energy is calculated using an Ewald method.¹⁰ A total charge density is defined by adding Gaussian charge distributions of the form

$$n^A(\mathbf{r}) = -\frac{Z_A}{(R_A^c)^3} \pi^{-3/2} \exp\left[-\left(\frac{\mathbf{r}-\mathbf{A}}{R_A^c}\right)^2\right] \quad (10)$$

to the electronic charge distribution $n(\mathbf{r})$. Therewith, Eq. (9) can be rewritten as

$$\begin{aligned} E[\mathbf{P}] = & \sum_{\mu\nu} P_{\mu\nu} \langle \varphi_{\mu}(\mathbf{r}) | -\frac{1}{2}\nabla^2 + V^{\text{ext}}(\mathbf{r}) | \varphi_{\nu}(\mathbf{r}) \rangle \\ & + 4\pi\Omega \sum_{\mathbf{G}} \frac{|n_{\text{tot}}(\mathbf{G})|^2}{G^2} + E_{\text{ovrl}} - E_{\text{self}} \\ & + \frac{\Omega}{N_1 N_2 N_3} \sum_{\mathbf{R}} F_{\text{XC}}[n_{\mathbf{G}}](\mathbf{R}), \end{aligned} \quad (11)$$

where the self and overlap terms

$$E_{\text{self}} = \sum_A \frac{1}{\sqrt{2\pi}} \frac{Z_A^2}{R_A^c}, \quad (12a)$$

$$E_{\text{ovrl}} = \sum_{A,B} \frac{Z_A Z_B}{|\mathbf{A}-\mathbf{B}|} \operatorname{erfc}\left[\frac{|\mathbf{A}-\mathbf{B}|}{\sqrt{R_A^{c2} + R_B^{c2}}}\right], \quad (12b)$$

as generated by these Gaussian distributions, have to be compensated. The double sum for E_{ovrl} runs over unique atom pairs and has to be extended, if necessary, beyond the minimum image convention. The electrostatic term also includes an interaction of the compensation charge with the electronic charge density that has to be subtracted from the external potential energy. The correction potential is of the form

$$V_{\text{core}}^A = \int d\mathbf{r}' \frac{n^A(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} = -\frac{Z_A}{|\mathbf{r}-\mathbf{A}|} \operatorname{erf}\left[\frac{|\mathbf{r}-\mathbf{A}|}{R_A^c}\right]. \quad (13)$$

The treatment of the electrostatic energy terms and the XC energy is the same as in PW codes.¹⁰ This means that the same methods that are used in those codes to adapt Eq. (11) for cluster boundary conditions can be used here. This includes analytic Green's function methods,¹⁰ the method of Eastwood and Brownrigg,¹¹ the methods of Martyna and Tuckerman,^{12,13} and wavelet approaches.^{14,15} Starting from $n(\mathbf{R})$, using Fourier transformations, we can calculate the combined potential for the electrostatic and XC energies. This includes the calculation of the gradient of the charge density needed for generalized gradient approximation (GGA) functionals. For non-local van der Waals functionals,¹⁶ we use the Fourier transform based algorithm of Román-Pérez and Soler.¹⁷ The combined potential calculated on the grid points $V^{\text{XC}}(\mathbf{R})$ is then the starting point to calculate the KS matrix elements

$$H_{\mu\nu}^{\text{XC}} = \frac{\Omega}{N_1 N_2 N_3} \sum_{\mathbf{R}} V^{\text{XC}}(\mathbf{R}) \varphi_{\mu}(\mathbf{R}) \varphi_{\nu}(\mathbf{R}). \quad (14)$$

This inverse mapping $V^{\text{XC}}(\mathbf{R}) \rightarrow H_{\mu\nu}^{\text{XC}}$ is using the same methods that have been used for the charge density. In this case, we can

achieve linear scaling with small prefactors. Furthermore, a large variety of different techniques to solve the eigenvalue problem for the now completed KS Hamilton matrix, so of which are linear scaling too, are described in detail in Sec. VIII.

A consistent calculation of nuclear forces within the GPW method can be done easily. Pulay terms, i.e., the dependence of the basis functions on nuclear positions, require the integration of potentials given on the real-space grid with derivatives of the basis functions.¹⁸ However, the basic routines work with Cartesian Gaussian functions and their derivatives are again functions of the same type, so the same routine can be used. Similarly, for XC functionals including the kinetic energy density, we can calculate $\tau(\mathbf{r})$ and the corresponding potential and matrix representation using the same mapping functions.

For the internal stress tensor

$$\Pi_{uv} = -\frac{1}{\Omega} \sum_s \frac{\partial E[\mathbf{P}]}{\partial \mathbf{h}_{us}} \mathbf{h}_{sv}^t, \quad (15)$$

we use again the Fourier transform framework for the E^{XC} terms and the simple virial for all pair forces. This applies to all Pulay terms, and only for the XC energy contributions from GGA functionals, special care is needed due to the cell dependent grid integration.^{19–21}

B. Dual-space pseudopotentials

The accuracy of the PW expansion of the electron density in the GPW method is controlled by the cutoff value E_{cut} restricting the maximal allowed value of $|\mathbf{G}|^2$. In the case of Gaussian basis sets, the cutoff needed to get a given accuracy is proportional to the largest exponent. As can be easily seen by inspecting common Gaussian basis sets for elements of different rows in the periodic table, the value of the largest exponent rapidly increases with atomic number. Therefore, the prefactor in GPW calculations will increase similarly. In order to avoid this, we can either resort to a pseudopotential description of inner shells or use a dual representation as described in Blöchl's projector augmented-wave method (PAW).²² The pseudopotentials used together with PW basis sets are constructed to generate nodeless atomic valence functions. Fully non-local forms are computationally more efficient and easier to implement. Dual-space pseudopotentials are of this form and are, because of their analytic form consistent of Gaussian functions, easily applied together with Gaussian basis sets.^{23–25}

The pseudopotentials are given in real-space as

$$V_{\text{loc}}^{\text{PP}}(r) = -\frac{Z_{\text{ion}}}{r} \text{erf}(\alpha^{\text{PP}} r) + \sum_{i=1}^4 C_i^{\text{PP}} \left(\sqrt{2} \alpha^{\text{PP}} r \right)^{2i-2} \times \exp\left[-(\alpha^{\text{PP}} r)^2\right], \text{ with } \alpha^{\text{PP}} = \frac{1}{\sqrt{2} r_{\text{loc}}^{\text{PP}}} \quad (16)$$

and a non-local part

$$V_{\text{nl}}^{\text{PP}}(\mathbf{r}, \mathbf{r}') = \sum_{lm} \sum_{ij} \langle \mathbf{r} | p_i^{lm} \rangle h_{ij}^l \langle p_j^{lm} | \mathbf{r}' \rangle, \quad (17)$$

where

$$\langle \mathbf{r} | p_i^{lm} \rangle = N_i^l Y^{lm}(\hat{\mathbf{r}}) r^{l+2i-2} \exp\left[-\frac{1}{2} \left(\frac{r}{r_l}\right)^2\right] \quad (18)$$

are Gaussian-type projectors resulting in a fully analytical formulation that requires only the definition of a small parameter set for each element. Moreover, the pseudopotentials are transferable and norm-conserving. The pseudopotential parameters are optimized with respect to the atomic all-electron wavefunction obtained from relativistic density functional calculations using a numerical atom code, which is also part of CP2K. A database with many pseudopotential parameter sets optimized for different XC potentials is available together with the distribution of the CP2K program.

C. Basis sets

The use of pseudopotentials in the GPW method also requires the use of correspondingly adapted basis sets. In principle, the same strategies that have been used to generate molecular or solid-state Gaussian basis sets could be used. It is always possible to generate specific basis sets for an application type, e.g., for metals or molecular crystals, but for ease of use, a general basis set is desirable. Such a general basis should fulfill the following requirements: high accuracy for smaller basis sets and a route for systematic improvements. One and the same basis set should perform in various environments from isolated molecules to condensed phase systems. Ideally, the basis sets should lead for all systems to well conditioned overlap matrices and be therefore well suited for linear scaling algorithms. To fulfill all the above requirements, generally contracted basis sets with shared exponents for all angular momentum states were proposed.²⁶ In particular, a full contraction over all primitive functions is used for both valence and polarization functions. The set of primitive functions includes diffuse functions with small exponents that are mandatory for the description of weak interactions. However, in contrast to the practice used in augmented basis sets, these primitive functions are always part of a contraction with tighter functions. Basis sets of this type were generated according to a recipe that includes global optimization of all parameters with respect to the total energy of a small set of reference molecules.²⁶ The target function was augmented with a penalty function that includes the overlap condition number. These basis sets of type *molo*pt have been created for the first two rows of the periodic table. They show good performance for molecular systems, liquids, and dense solids. The results for the delta test are shown in Fig. 18. The basis sets have seven primitive functions with a smallest exponent of ≈ 0.047 for oxygen. This is very similar to the smallest exponent found in augmented basis sets of the correlation consistent type, e.g., ≈ 0.060 for aug-cc-pVTZ. The performance of the grid mapping routines depends strongly on this most diffuse function in the basis sets. We have therefore optimized a second set of basis sets of the *molo*pt type, where the number of primitives has been reduced (e.g., 5 for first row elements), which also leads to less diffuse functions. The smallest exponent for oxygen in this basis set is now ≈ 0.162 . These basis sets still show good performance in many different environments and are especially suited for all types of condensed matter systems (e.g., see the delta test results in Sec. XV D). The reduction of Gaussian primitives and the removal of very diffuse functions lead to a 10-fold time reduction for the mapping routines for liquid water calculations using default settings.

D. Local density fitting approach

Within the GPW approach, the mapping of the electronic density on the grid is often the time determining step. With an

appropriate screening, this is a linear scaling step with a prefactor determined by the number of overlapping basis functions. Especially in condensed phase systems, the number of atom pairs that have to be included can be very large. For such cases, it can be beneficial to add an intermediary step in the density mapping. In this step, the charge density is approximated by another auxiliary Gaussian basis. The expansion coefficients are determined using the local density fitting approach by Baerends *et al.*²⁷ They introduced a local metric, where the electron density is decomposed into pair-atomic densities, which are approximated as a linear combination of auxiliary functions localized at atoms A and B. The expansion coefficients are obtained by employing an overlap metric. This local resolution-of-the-identity (LRI) method combined with the GPW method is available in CP2K as the LRIGPW approach.²⁸ For details on how to set up such a calculation, see Ref. 29.

The atomic pair densities n^{AB} are approximated by an expansion in a set of Gaussian fit functions $f(\mathbf{r})$ centered at atoms A and B, respectively. The expansion coefficients are obtained for each pair AB by fitting the exact density while keeping the number of electrons fixed. This leads to a set of linear equations that can be solved easily. The total density is then represented by the sum of coefficients of all pair expansions on the individual atoms. The total density is now presented as a sum over the number of atoms, whereas in the GPW method, we have a sum over pairs. In the case of 64 water molecules in a periodic box, this means that the fitted density is mapped on the grid by 192 atom terms rather than $\approx 200\,000$ atom pairs.

LRIGPW requires the calculation of two- and three-index overlap integrals that is computationally demanding for large auxiliary basis sets. To increase the efficiency of the LRIGPW implementation, we developed an integral scheme based on solid harmonic Gaussian functions,³⁰ which is superior to the widely used Cartesian Gaussian-based methods.

An additional increase in efficiency can be achieved by recognizing that most of the electron density is covered by a very small number of atom pair densities. The large part of more distant pairs can be approximated by an expansion on a single atom. By using a distance criterion and a switching function, a method with a smooth potential energy is created. The single atom expansion reduces memory requirements and computational time considerably. In the above water box example, about 99% of the pairs can be treated as distant pairs.

E. Gaussian-augmented plane wave approach

An alternative to pseudopotentials, or a method to allow for smaller cutoffs in pseudopotential calculations, is provided by the Gaussian-augmented plane wave (GAPW) approach.^{31,32} The GAPW method uses a dual representation of the electronic density, where the usual expansion of the density using the density matrix $P_{\alpha\beta}$ is replaced in the calculation of the Coulomb and XC energy by

$$n(\mathbf{r}) = \tilde{n}(\mathbf{r}) + \sum_A n_A(\mathbf{r}) - \sum_A \tilde{n}_A(\mathbf{r}). \quad (19)$$

The densities $\tilde{n}(\mathbf{r})$, $n_A(\mathbf{r})$, and $\tilde{n}_A(\mathbf{r})$ are expanded in plane waves and products of primitive Gaussians centered on atom A, respectively, i.e.,

$$\tilde{n}(\mathbf{r}) = \frac{1}{\Omega} \sum_{\mathbf{G}} \tilde{n}(\mathbf{G}) e^{i\mathbf{G}\cdot\mathbf{r}}, \quad (20a)$$

$$n_A(\mathbf{r}) = \sum_{mn \in A} P_{mn}^A g_m(\mathbf{r}) g_n^*(\mathbf{r}), \quad (20b)$$

$$\tilde{n}_A(\mathbf{r}) = \sum_{mn \in A} \tilde{P}_{mn}^A g_m(\mathbf{r}) g_n^*(\mathbf{r}). \quad (20c)$$

In Eq. (20a), $\tilde{n}(\mathbf{G})$ are the Fourier coefficients of the soft density, as obtained in the GPW method by keeping in the expansion of the contracted Gaussians only those primitives with exponents smaller than a given threshold. The expansion coefficients P_{mn}^A and \tilde{P}_{mn}^A are also functions of the density matrix $P_{\alpha\beta}$ and can be calculated efficiently. The separation of the density from Eq. (19) is borrowed from the PAW approach.²² Its special form allows the separation of the smooth parts, characteristic of the interatomic regions, from the quickly varying parts close to the atoms while still expanding integrals over all space. The sum of the contributions in Eq. (19) gives the correct full density if the following conditions are fulfilled:

$$n(\mathbf{r}) = n_A(\mathbf{r}), \quad \tilde{n}(\mathbf{r}) = \tilde{n}_A(\mathbf{r}) \quad \text{close to atom A}, \quad (21a)$$

$$n(\mathbf{r}) = \tilde{n}(\mathbf{r}), \quad n_A(\mathbf{r}) = \tilde{n}_A(\mathbf{r}) \quad \text{far from atom A}. \quad (21b)$$

The first conditions are exactly satisfied only in the limit of a complete basis set. However, the approximation introduced in the construction of the local densities can be systematically improved by choosing larger basis sets.

For semi-local XC functionals such as the local density approximation (LDA), GGA, or meta functionals using the kinetic energy density, the XC energy can be simply written as

$$E_{\text{GAPW}}^{\text{XC}}[n] = E^{\text{XC}}[\tilde{n}] + \sum_A E^{\text{XC}}[n_A] - \sum_A E^{\text{XC}}[\tilde{n}_A]. \quad (22)$$

The first term is calculated on the real-space grid defined by the PW expansion, and the other two are efficiently and accurately calculated using atom centered meshes.

Due to the non-local character of the Coulomb operator, the decomposition for the electrostatic energy is more complex. In order to distinguish between local and global terms, we need to introduce atom-dependent screening densities n_A^0 that generate the same multipole expansion Q_A^{lm} as the local density $n_A - \tilde{n}_A + n_A^Z$, where n_A^Z is the nuclear charge of atom A, i.e.,

$$n_A^0(\mathbf{r}) = \sum_{lm} Q_A^{lm} g_A^{lm}(\mathbf{r}). \quad (23)$$

The normalized primitive Gaussians $g_A^{lm}(\mathbf{r})$ are defined with an exponent such that they are localized within an atomic region. Since the sum of local densities $n_A - \tilde{n}_A + n_A^Z - n_A^0$ has vanishing multiple moments, it does not interact with charges outside the localization region, and the corresponding energy contribution can be calculated by one-center integrals. The final form of the Coulomb energy in the GAPW method then reads as

$$E_{\text{GAPW}}^{\text{C}}[n + n^Z] = E^{\text{H}}[\tilde{n} + n^0] + \sum_A E^{\text{H}}[n_A + n_A^Z] - \sum_A E^{\text{H}}[\tilde{n}_A + n_A^0], \quad (24)$$

where n^0 is summed over all atomic contributions and $E^H[n]$ denotes the Coulomb energy of a charge distribution n . The first term in Eq. (24) can be efficiently calculated using fast Fourier transform (FFT) methods using the GPW framework. The one-centered terms are calculated on radial atomic grids.

The special form of the GAPW energy functional involves several additional approximations in addition to a GPW calculation. The accuracy of the local expansion of the density is controlled by the flexibility of the product basis of primitive Gaussians. As we fix this basis to be the primitive Gaussians present in the original basis, we cannot independently vary the accuracy of the expansion. Therefore, we have to consider this approximation as inherent to the primary basis used.

With the GAPW method, it is possible to calculate materials properties that depend on the core electrons. This has been used for the simulation of the x-ray scattering in liquid water.³³ X-ray absorption (XAS) spectra are calculated using the transition potential method.^{34–40} Several nuclear and electronic magnetic properties such as nuclear magnetic resonance (NMR) chemical shift and electric field gradient, electron paramagnetic resonance (EPR) hyperfine coupling, and g-tensors are also available.^{41–44}

III. HARTREE-FOCK AND HYBRID DENSITY FUNCTIONAL THEORY METHODS

Even though semi-local DFT is a cornerstone of much of condensed phase electronic structure modeling,⁴⁵ it is also recognized that going beyond GGA-based DFT is necessary to improve the accuracy and reliability of electronic structure methods.⁴⁶ One path forward is to augment DFT by elements of wavefunction theory⁴⁷ or to adopt wavefunction theory itself.⁴⁸ This is the approach taken in successful hybrid XC functionals such as B3LYP or Heyd-Scuseria-Ernzerhof (HSE),^{49,50} where part of the exchange functional is replaced by exact Hartree-Fock exchange (HFX). The capability to compute HFX was introduced in CP2K by Guidon *et al.*^{51–53} The aim at that time was to enable the use of hybrid XC functionals for condensed phase calculations, of relatively large, disordered systems, in the context of AIMD simulations. This objective motivated a number of implementation choices and developments that will be described in the following. The capability was particularly important to make progress in the field of first principles electro-chemistry^{54,55} but is also the foundation for the correlated wavefunction (CW) methods such as MP2, RPA, and GW that are available in CP2K and will be described in Sec. IV.

In the periodic case, HFX can be computed as

$$E_X^{\text{PBC}} = -\frac{1}{2N_k} \sum_{ij} \sum_{\mathbf{k}\mathbf{k}'} \iint \psi_i^{\mathbf{k}}(r_1) \psi_j^{\mathbf{k}'}(r_1) \times g(|r_1 - r_2|) \psi_i^{\mathbf{k}}(r_2) \psi_j^{\mathbf{k}'}(r_2) d^3r_1 d^3r_2, \quad (25)$$

where an explicit sum over the k-points is retained and a generalized Coulomb operator $g(|r_1 - r_2|)$ is employed. The k-point sum is important as at least for the standard Coulomb operator $1/r$, the term for $\mathbf{k} = \mathbf{k}' = \mathbf{0}$ (the so-called Γ -point) is singular, although the full sum approximates an integrable expression. If the operator is short-ranged or screened, the Γ -point term is well-behaved. CP2K computes HFX at the Γ -point only and employs a localized atomic

basis set, using an expression where the sum is explicit over the indices of the localized basis ($\lambda\sigma\mu\nu$) as well as the image cells (\mathbf{abc}), thus

$$E_X^{\Gamma} = -\frac{1}{2} \sum_{\lambda\sigma\mu\nu} P^{\mu\sigma} P^{\nu\lambda} \sum_{\mathbf{abc}} (\mu\nu^{\mathbf{a}} | \lambda^{\mathbf{b}} \sigma^{\mathbf{b+c}})_{\mathbf{g}}. \quad (26)$$

For an operator with a finite range, the sum over the image cells will terminate. This expression was employed to perform hybrid DFT-based AIMD simulations of liquid water with CP2K.⁵¹ Several techniques have been introduced to reduce the computational cost. First, screening based on the overlap of basis functions is employed to reduce the scaling of the calculation from $O(N^4)$ to $O(N^2)$. This does not require any assumptions on the sparsity of the density matrix, nor the range of the operator, and makes HFX feasible for fairly large systems. Second, the HFX implementation in CP2K is optimized for “in-core” calculations, where the four center integrals are computed (analytically) only once at the beginning of the SCF procedure, stored in main memory, and reused afterward. This is particularly useful in the condensed phase as the sum over all image cells multiplies the cost of evaluating the integral, relative to gas phase calculations. To store all computed integrals, the code has been very effectively parallelized using Message Passing Interface (MPI) and Open Multi-Processing (OpenMP), yielding super-linear speed-ups as long as added hardware resources provide additional memory to store all integrals. Furthermore, a specialized compression algorithm is used to store each integral with just as many bits as needed to retain the target accuracy. Third, a multiple-time-step (MTS) algorithm (see Sec. X D) is employed to evaluate HFX energies only every few time steps during an AIMD simulation, assuming that the difference between the potential energy surface of a GGA and a hybrid XC functional is slowly varying with time. This technique has found reuse in the correlated wavefunction simulations described in Sec. IV.

In Ref. 52, the initial implementation was revisited, in particular, to be able to robustly compute HFX at the Γ -point for the case where the operator in the exchange term is $\frac{1}{r}$, and not a screened operator as, e.g., in HSE.^{56,57} The solution is to truncate the operator (not any of the image cell sums), with a truncation radius R_C that grows with the cell size. The advantage of this approach is that the screening of all terms in Eq. (26) can be performed rigorously and that the approach is stable for a proper choice of screening threshold, also in the condensed phase with good quality (but non-singular) basis sets. The value of R_C that yields convergence is system dependent, and large values of R_C might require the user to explicitly consider multiple unit cells for the simulation cell. Note that the HFX energy converges exponentially with R_C for typical insulating systems and that the same approach was used previously to accelerate k-point convergence.⁵⁸ In Ref. 59, it was demonstrated that two different codes (CP2K and Gaussian), with very different implementations of HFX, could reach micro-Hartree agreement for the value of the HF total energy of the LiH crystal. In Ref. 52, a suitable GGA-type exchange function was derived that can be used as a long range correction (LRC) together with the truncated operator. This correction functional, in the spirit of the exchange functional employed in HSE, effectively allows for model chemistries that employ very short range exchange (e.g., ≈ 2 Å) only.

Important for the many applications of HFX with CP2K is the auxiliary density matrix method (ADMM), introduced in Ref. 53.

This method reduces the cost of HFX significantly, often bringing it to within a few times the cost of conventional GGA-based DFT, by addressing the unfavorable scaling of the computational cost of Eq. (26) with respect to basis set size. The key ingredient of the ADMM method is the use of an auxiliary density matrix $\hat{\mathbf{P}}$, which approximates the original \mathbf{P} , for which the HFX energy is more cost-effective to compute,

$$\begin{aligned} E_X^{\text{HFX}}[\mathbf{P}] &= E_X^{\text{HFX}}[\hat{\mathbf{P}}] + (E_X^{\text{HFX}}[\mathbf{P}] - E_X^{\text{HFX}}[\hat{\mathbf{P}}]) \\ &\approx E_X^{\text{HFX}}[\hat{\mathbf{P}}] + (E_X^{\text{DFT}}[\mathbf{P}] - E_X^{\text{DFT}}[\hat{\mathbf{P}}]). \end{aligned} \quad (27)$$

Effectively, $E_X^{\text{HFX}}[\mathbf{P}]$ is replaced with computationally more efficient $E_X^{\text{HFX}}[\hat{\mathbf{P}}]$, and the difference between the two is corrected approximately with a GGA-style exchange functional. Commonly, the auxiliary $\hat{\mathbf{P}}$ is obtained by projecting the density matrix \mathbf{P} using a smaller, auxiliary basis. This approximation, including the projection, can be implemented fully self-consistently. In Ref. 60, the efficiency of the ADMM method was demonstrated by computing over 300 ps of AIMD trajectory for systems containing 160 water molecules and by computing spin densities for a solvated metalloprotein system containing ~3000 atoms.⁵³

IV. BEYOND HARTREE-FOCK METHODS

In addition, so-called post-Hartree-Fock methods that are even more accurate than the just describe hybrid-DFT approach are also available within CP2K.

A. Second-order Møller-Plesset perturbation theory

Second-order Møller-Plesset perturbation theory (MP2) is the simplest *ab initio* correlated wavefunction method⁶¹ applied to the Hartree-Fock reference and able to capture most of the dynamic electron correlation.⁶² In the DFT framework, the MP2 formalism gives rise to the doubly hybrid XC functionals.⁶³ In the spin-orbital basis, the MP2 correlation energy is given by

$$\begin{aligned} E^{(2)} &= \frac{1}{2} \left\{ \sum_{ij,ab} \frac{(ia|jb)[(ia|jb) - (ib|ja)]}{\Delta_{ij}^{ab}} \right. \\ &\quad \left. + \sum_{\bar{i}\bar{j},\bar{a}\bar{b}} \frac{(\bar{i}\bar{a}|\bar{j}\bar{b})[(\bar{i}\bar{a}|\bar{j}\bar{b}) - (\bar{i}\bar{b}|\bar{j}\bar{a})]}{\Delta_{\bar{i}\bar{j}}^{\bar{a}\bar{b}}} \right\} - \sum_{\bar{i}\bar{j},\bar{a}\bar{b}} \frac{(\bar{i}\bar{a}|\bar{j}\bar{b})}{\Delta_{\bar{i}\bar{j}}^{\bar{a}\bar{b}}}, \end{aligned} \quad (28)$$

where i, j, \dots run over occupied spin-orbitals, a, b, \dots run over virtual spin-orbitals (indexes without bar stand for α -spin-orbitals and indexes with bar stand for β -spin-orbitals), $\Delta_{ij}^{ab} = \epsilon_a + \epsilon_b - \epsilon_i - \epsilon_j$ (ϵ_a and ϵ_i are orbital energies), and $(ia|jb)$ are electron repulsion integrals in the Mulliken notation.

In a canonical MP2 energy algorithm, the time limiting step is the computation of the $(ia|jb)$ integrals obtained from the electron repulsion integrals over AOs ($\mu\nu|\lambda\sigma$) via four consecutive index integral transformations. The application of the resolution-of-identity (RI) approximation to MP2,⁶⁴ which consists of replacing integrals $(ia|jb)$ by its approximated counterparts $(ia|jb)_{\text{RI}}$, is given by

$$(ia|jb)_{\text{RI}} = \sum_P B_P^{ia} B_P^{jb}, \quad (29)$$

with

$$B_P^{ia} = \sum_R (ia|R) L_{PR}^{-1}, \quad (30)$$

where P, R, \dots (the total number of them is N_a) are auxiliary basis functions and \mathbf{L} are two-center integrals over them. The RI-MP2 method is also scaling $O(N^5)$ with a lower prefactor: the main reason for the speed-up in RI-MP2 lies in the strongly reduced integral computation cost.

As the MP2 is non-variational with respect to wavefunction parameters, analytical expressions for geometric energy derivatives of RI-MP2 energies are complicated, since its calculation requires the solution of Z-vector equations.⁶⁵

1. Scaled opposite-spin MP2

The scaled opposite-spin MP2 (SOS-MP2) method is a simplified variant of MP2.⁶⁶ Starting from Eq. (28), we neglect the same spin term in curly brackets and scale the remaining opposite spin term to account for the introduced error. We can rewrite the energy term with the RI approximation and the Laplace transform,

$$\frac{1}{x} = \int_0^\infty dt e^{-xt}. \quad (31)$$

When we exploit a numerical integration, we obtain the working equation for SOS-MP2, which reads as

$$E^{\text{SOS-MP2}} = - \sum_q \sum_{PQ} w_q Q_{PQ}^\alpha(t_q) Q_{QP}^\beta(t_q) \quad (32)$$

with

$$Q_{PQ}^\alpha(t_q) = \sum_{ia} B_P^{ia} B_Q^{ia} e^{(\epsilon_a - \epsilon_i)t_q} \quad (33)$$

and similarly for the beta spin. The integration weights w_q and abscissa t_q are determined by a minimax procedure.⁶⁷ In practice, ≈ 7 quadrature points are needed for $\mu\text{hartree}$ accuracy.⁶⁶

The most expensive step is the calculation of the matrix elements Q_{PQ}^α , which scales like $O(N^4)$. Due to similarities with the random phase approximation (RPA), we will discuss the implementation of this method in Sec. IV B.

2. Implementation

CP2K features Γ -point implementations of canonical MP2, RI-MP2, Laplace-transformed MP2, and SOS-MP2 energies.^{68,69} For RI-MP2, analytical gradients and stress tensors are available,⁷⁰ for both closed and open electronic shells.⁷¹ Two- and three-center integrals can be calculated by the GPW method or analytically.

The implementation is massively parallel, directly profits from the availability of sparse matrix algebra, and allows for graphics processing unit (GPU) acceleration of large matrix multiplies. The evaluation of the gradients of the RI-MP2 energy can be performed within a few minutes for systems containing hundreds of atoms and thousands of basis functions on thousands of central processing unit (CPU) cores, allowing for MP2-based structure relaxation and even AIMD simulations on HPC facilities. The cost of the gradient calculation is 4–5 times larger than the energy evaluation, and open-shell MP2 calculation is typically 3–4 times more expensive than the closed-shell calculation.

3. Applications

The RI-MP2 implementation, which is the computationally most efficient MP2 variant available in CP2K, has been successfully applied to study a number of aqueous systems. In fact, *ab initio* Monte Carlo (MC) and AIMD simulations of bulk liquid water (with simulation cells containing 64 water molecules) predicted the correct density, structure, and IR spectrum.^{72–74} Other applications include structure refinements of ice XV,⁷⁵ AIMD simulations of the bulk hydrated electron (with simulation cells containing 47 water molecules),⁷⁶ as well as the first AIMD simulation of a radical in the condensed phase using wavefunction theory.

B. Random phase approximation correlation energy method

Total energy methods based on the RPA correlation energy have emerged in the recent years as promising approaches to include non-local dynamical electron correlation effects at the fifth rung on the Jacob's ladder of density functional approximations.⁷⁷ In this context, there are numerous ways to express the RPA correlation energy depending on the theoretical framework and approximations employed to derive the working equations.^{78–96} Our implementation uses the approach introduced by Eshuis *et al.*,⁹⁷ which can be referred to as based on the dielectric matrix formulation, involving the numerical integral over the frequency of a logarithmic expression including the dynamical dielectric function, expressed in a Gaussian RI auxiliary basis. Within this approach, the direct-RPA (sometimes referred to as dRPA) correlation energy, which is a RPA excluding exchange contributions,⁹⁷ is formulated as a frequency integral

$$E_c^{\text{RPA}} = \frac{1}{2} \int_{-\infty}^{+\infty} \frac{d\omega}{2\pi} \text{Tr}(\ln(1 + \mathbf{Q}(\omega)) - \mathbf{Q}(\omega)), \quad (34)$$

with the frequency dependent matrix $\mathbf{Q}(\omega)$, expressed in the RI basis, determined by

$$Q_{PQ}(\omega) = 2 \sum_i^a \sum_a^v B_p^{ia} \frac{\epsilon_a - \epsilon_i}{\omega^2 + (\epsilon_a - \epsilon_i)^2} B_Q^{ia}. \quad (35)$$

For a given ω , the computation of the integrand function in Eq. (34) and using Eq. (35) requires $O(N^4)$ operations. The integral of Eq. (34) can be efficiently calculated by a minimax quadrature requiring only ≈ 10 integration points to achieve μ_{hartree} accuracy.^{98,99} Thus, the introduction of the RI approximation and the frequency integration techniques for computing E_c^{RPA} leads to a computational cost of $O(N^4 N_q)$ and $O(N^3)$ storage, where N_q is the number of quadrature points.⁶⁹

We note here that the conventional RPA correlation energy methods in CP2K are based on the exact exchange (EXX) and RPA correlation energy formalism (EXX/RPA), which has extensively been applied to a large variety of systems including molecules, systems with reduced dimensionality, and solids.^{75,78–81,83,84,86–89,100–111} Within the framework of the EXX/RPA formalism, the total energy is given as

$$\begin{aligned} E_{\text{tot}}^{\text{EXX/RPA}} &= E_{\text{tot}}^{\text{HF}} + E_c^{\text{RPA}} \\ &= (E_{\text{tot}}^{\text{DFT}} - E_{\text{XC}}^{\text{DFT}}) + E_X^{\text{EXX}} + E_c^{\text{RPA}}, \end{aligned} \quad (36)$$

where the right-hand side (RHS) terms of the last equation are the DFT total energy, the DFT xc energy, the EXX energy, and the (direct) RPA correlation energy, respectively. The sum of the first three terms is identical to the Hartree–Fock energy as calculated employing DFT orbitals, which is usually denoted as HF@DFT. The last term corresponds to the RPA correlation energy as computed using DFT orbitals and orbital energies and is often referred to as RPA@DFT. The calculation of $E_{\text{tot}}^{\text{EXX/RPA}}$ thus requires a ground state calculation with a given DFT functional, followed by an EXX energy evaluation and a RPA correlation energy evaluation employing DFT ground state wavefunctions and orbital energies.

1. Implementation of the quartic scaling RPA and SOS-MP2 methods

We summarize here the implementation in CP2K of the quartic scaling computation of the RPA and SOS-MP2 correlation energies. The reason to describe the two implementations here is due to the fact that the two approaches share several components. In fact, it can be shown that the direct MP2 energy can be obtained by truncating at the first non-vanishing term of the Taylor expansion to compute the logarithm in Eq. (34) and integrating over frequencies.⁹⁷

After the three center RI integral matrix B_p^{ia} is made available (via, i.e., RI-GPW⁶⁹), the key component of both methods is the evaluation of the frequency dependent $Q_{PQ}(\omega)$ for the RPA and the τ dependent $Q_{PQ}(\tau)$ for the Laplace transformed SOS-MP2 method (see Sec. IV A 1). The matrices $Q_{PQ}(\omega)$ and $Q_{PQ}(\tau)$ are given by the contractions in Eqs. (35) and (33), respectively. Their computation entails, as a basic algorithmic motif, a large distributed matrix multiplication between tall and skinny matrices for each quadrature point. Fortunately, the required operations at each quadrature point are independent of each other. The parallel implementation in CP2K exploits this fact by distributing the workload for the evaluation of $Q_{PQ}(\omega)$ and $Q_{PQ}(\tau)$ over pools of processes, where each pool is working independently on a subset of quadrature points. Furthermore, the operations necessary for each quadrature point are performed in parallel within all members of the pool. In this way, the $O(N^4)$ bottleneck of the computation displays an embarrassingly parallel distribution of the workload and, in fact, it shows excellent parallel scalability to several thousand nodes.^{69,98} Additionally, since at the pool level, the distributed matrix multiplication employs the widely adopted data layout of the parallel BLAS library, minimal modifications are required to exploit accelerators [such as graphics processing units (GPUs) and field-programmable gate arrays (FPGAs); see Sec. XV A for details], as interfaces to the corresponding accelerated libraries are made available.⁹⁸

Finally, the main difference between RPA and SOS-MP2 is the postprocessing. After the contraction step to obtain the Q_{PQ} matrix, the sum in Eq. (32) is performed with computational costs of $O(N_a^2 N_q)$ for SOS-MP2 instead of $O(N_q^3 N_q)$, which is associated with the evaluation of the matrix logarithm of Eq. (34) for the RPA (in CP2K, this operation is performed using the identity $\text{Tr}[\ln \mathbf{A}] = \ln(\det[\mathbf{A}])$, where a Cholesky decomposition is used to efficiently calculate the matrix determinant). Therefore, the computational costs of the quartic-scaling RPA and SOS-MP2 are the same for large systems, assuming that the same number of quadrature points is used for the numerical integration.

2. Cubic scaling RPA and SOS-MP2 method

The scaling of RPA and SOS-MP2 can be reduced from $O(N^4)$ to $O(N^3)$ or even better by alternative analytical formulations of the methods.^{112,113} Here, we describe the CP2K-specific cubic scaling RPA/SOS-MP2 implementation and demonstrate the applicability to systems containing thousands of atoms.

For cubic scaling RPA calculations, the matrix $Q_{PQ}(\omega)$ from Eq. (35) is transformed to imaginary-time $Q_{PQ}(\tau)$ ⁹⁹ as it is already present in SOS-MP2. The tensor B_p^{ia} is transformed from occupied-virtual MO pairs ia to pairs $\mu\nu$ of AO basis set functions. This decouples the sum over occupied and virtual orbitals and thereby reduces the formal scaling from quartic to cubic. Further requirements for a cubic scaling behavior are the use of localized atomic Gaussian basis functions and the localized overlap RI metric such that the occurring 3-center integrals are sparse. A sparse representation of the density matrix is not a requirement for our cubic scaling implementation, but it reduces the effective scaling of the usually dominant $O(N^2)$ sparse tensor contraction steps to $O(N)$.¹¹²

The operations performed for the evaluation of $Q_{PQ}(\tau)$ are generally speaking contractions of sparse tensors of ranks 2 and 3 – starting from the 3-center overlap integrals and the density matrix. Consequently, sparse linear algebra is the key to good performance, as opposed to the quartic scaling implementation that relies mostly on parallel BLAS for dense matrix operations.

The cubic scaling RPA/SOS-MP2 implementation is based on the distributed block compressed sparse row (DBCSR) library,¹¹⁴ which is described in detail in Sec. XIII and was originally co-developed with CP2K, to enable linear scaling DFT.¹¹⁵ The library was extended with a tensor API in a recent effort to make it more easily applicable to algorithms involving contractions of large sparse multi-dimensional tensors. Block-sparse tensors are internally represented as DBCSR matrices, whereas tensor contractions are mapped to sparse matrix–matrix multiplications. An in-between tall-and-skinny matrix layer reduces memory requirements for storage and reduces communication costs for multiplications by splitting the largest matrix dimension and running a simplified variant of the CARMA algorithm.¹¹⁶ The tensor extension of the DBCSR library leads to significant improvements in terms of performance and usability compared to the initial implementation of the cubic scaling RPA.¹¹³

In Fig. 1, we compare the computational costs of the quartic and the cubic scaling RPA energy evaluation for periodic water systems of different sizes. All calculations use the RI together with the overlap metric and a high-quality cc-TZVP primary basis with a matching RI basis. The neglect of small tensor elements in the $O(N^3)$ implementation is controlled by a filtering threshold parameter. This parameter has been chosen such that the relative error introduced in the RPA energy is below 0.01%. The favorable effective scaling of $O(N^{1.8})$ in the cubic scaling implementation leads to better absolute timings for systems of 100 or more water molecules. At 256 water molecules, the cubic scaling RPA outperforms the quartic scaling method by one order of magnitude.

The observed scaling is better than cubic in this example because the $O(N^3)$ scaling steps have a small prefactor and would start to dominate in systems larger than the ones presented here—they make up for around 20% of the execution time for the largest system. The dominant sparse tensor contractions are quadratic

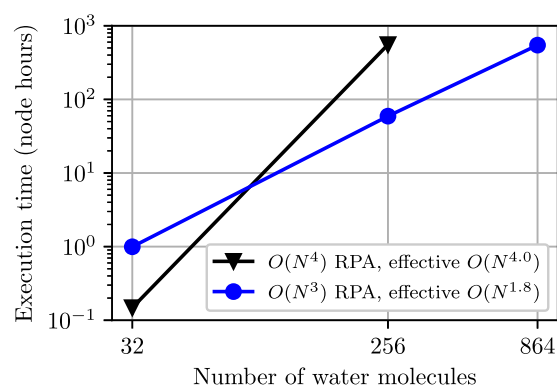


FIG. 1. Comparison of the timings for the calculation of the RPA correlation energy using the quartic- and the cubic-scaling implementation on a CRAY XC50 machine with 12 cores per node (flat MPI). The system sizes are $n \times n \times n$ supercells (with $n = 1, 2, 3$) of a unit cell with 32 water molecules and a density of 1 g/cm³. The intersection point where the cubic scaling methods become favorable is at ~90 water molecules. The largest system tested with the cubic scaling RPA contains 864 water molecules (6912 electrons, 49 248 primary basis functions, and 117 504 RI basis functions) and was calculated on 256 compute nodes (3072 cores). The largest tensor of size 117 504 × 49 248 × 49 248 has an occupancy below 0.2%.

scaling, closely matching the observed scaling of $O(N^{1.8})$. It is important to mention that the density matrices are not yet becoming sparse for these system sizes. Lower dimensional systems with large extent in one or two dimensions have an even more favorable scaling regime of $O(N)$ since the onset of sparse density matrices occurs at smaller system sizes.

All aspects of the comparison discussed here also apply to SOS-MP2 because it shares the algorithm and implementation of the dominant computational steps with the cubic scaling RPA method. In general, the exact gain of the cubic scaling RPA/SOS-MP2 scheme depends on the specifics of the applied basis sets (locality and size). The effective scaling, however, is $O(N^3)$ or better for all systems, irrespective of whether the density matrix has a sparse representation, thus extending the applicability of the RPA to large systems containing thousands of atoms.

C. Ionization potentials and electron affinities from GW

The GW approach, which allows us to approximately calculate the self-energy of a many-body system of electrons, has become a popular tool in theoretical spectroscopy to predict electron removal and addition energies as measured by direct and inverse photoelectron spectroscopy, respectively [see Figs. 2(a) and 2(b)].^{117–120} In direct photoemission, the electron is ejected from orbital ψ_n to the vacuum level by irradiating the sample with visible, ultraviolet, or X-rays, whereas in the inverse photoemission process, an injected electron undergoes a radiative transition into an unoccupied state.

The GW approximation has been applied to a wide range of materials including two-dimensional systems, surfaces, and molecules. A summary of applications and an introduction to the many-body theory behind GW and practical aspects can be found in a recent review article.¹²⁰ The electron removal energies are referred

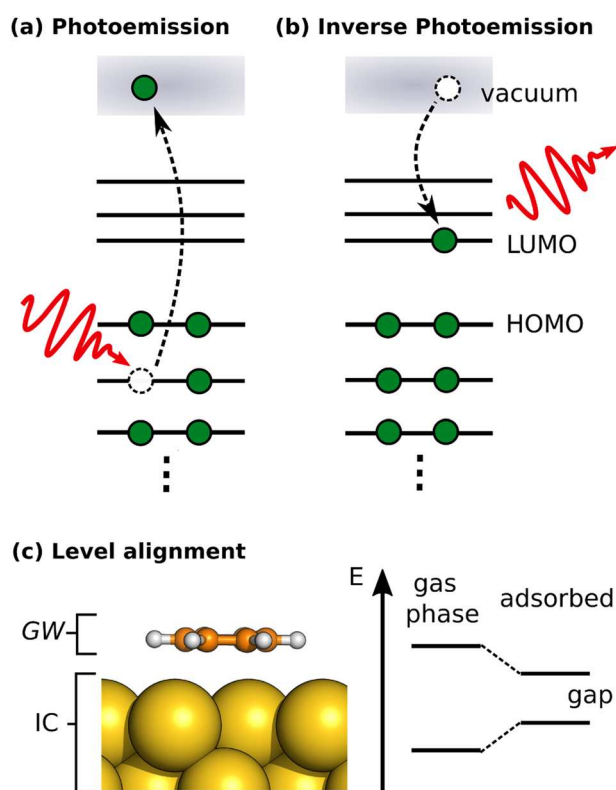


FIG. 2. Properties predicted by the GW method: Ionization potentials and electron affinities as measured by (a) photoemission and (b) inverse photoemission spectroscopy, respectively. (c) Level alignment of the HOMO and LUMO upon adsorption of a molecule at metallic surfaces, which are accounted for in CP2K by an IC model avoiding the explicit inclusion of the metal in the GW calculation.

to as ionization potentials (IPs) and the negative of the electron addition energies are referred to as electron affinities (EAs); see Ref. 120 for details on the sign convention. The GW method yields a set of energies $\{\epsilon_n\}$ for all occupied and unoccupied orbitals $\{\psi_n\}$. ϵ_n can be directly related to the IP for occupied states and to the EA for the lowest unoccupied state (LUMO). Hence,

$$\text{IP}_n = -\epsilon_n, \quad n \in \text{occ} \quad \text{and} \quad \text{EA} = -\epsilon_{\text{LUMO}}. \quad (37)$$

The difference between the IP of the highest occupied state (HOMO) and the EA is called the fundamental gap, a critical parameter for charge transport in, e.g., organic semiconductors.¹²¹ It should be noted that the fundamental HOMO–LUMO gap does not correspond to the first optical excitation energy (also called optical gap) that is typically smaller than the fundamental gap due to the electron–hole binding energy.¹²² For solids, the Bloch functions $\psi_{n\mathbf{k}}$ carry an additional quantum number \mathbf{k} in the first Brillouin zone and give rise to a band structure $\epsilon_{n\mathbf{k}}$.^{118,123} From the band structure, we can determine the bandgap, which is the solid-state equivalent to the HOMO–LUMO gap. Unlike for molecules, an angle-resolved photoemission experiment is required to resolve the \mathbf{k} -dependence.

For GW, mean absolute deviations of less than 0.2 eV from the higher-level coupled-cluster singles and doubles plus perturbative triples [CCSD(T)] reference have been reported for IP_{HOMO} and EA.^{124,125} The deviation from the experimental reference can be even reduced to <0.1 eV when including also vibrational effects.¹²⁶ On the contrary, DFT eigenvalues ϵ_n^{DFT} fail to reproduce spectroscopic properties. Relating ϵ_n^{DFT} to the IPs and EA as in Eq. (37) is conceptually only valid for the HOMO level.¹²⁷ Besides the conceptual issue, IPs from DFT eigenvalues are underestimated by several eV compared to experimental IPs due to the self-interaction error in GGA and LDA functionals yielding far too small HOMO–LUMO gaps.^{128,129} Hybrid XC functionals can improve the agreement with experiment, but the amount of exact exchange can strongly influence ϵ_n^{DFT} in an arbitrary way.

The most common GW scheme is the G_0W_0 approximation, where the GW calculation is performed non-self-consistently on top of an underlying DFT calculation. In G_0W_0 , the MOs from DFT ψ_n^{DFT} are employed and the DFT eigenvalues are corrected by replacing the incorrect XC contribution $\langle \psi_n^{\text{DFT}} | v^{\text{xc}} | \psi_n^{\text{DFT}} \rangle$ by the more accurate energy-dependent XC self-energy $\langle \psi_n^{\text{DFT}} | \Sigma(\epsilon) | \psi_n^{\text{DFT}} \rangle$, i.e.,

$$\epsilon_n = \epsilon_n^{\text{DFT}} + \langle \psi_n^{\text{DFT}} | \Sigma(\epsilon_n) - v^{\text{xc}} | \psi_n^{\text{DFT}} \rangle. \quad (38)$$

The self-energy is computed from the Green's function G and the screened Coulomb interaction W , $\Sigma = GW$, hence the name of the GW approximation.^{119,120}

The G_0W_0 implementation in CP2K works routinely for isolated molecules,¹³⁰ although first attempts have been made to extend it for periodic systems.^{76,131,132} The standard G_0W_0 implementation is optimized for computing valence orbital energies ϵ_n (e.g., up to 10 eV below the HOMO and 10 eV above the LUMO).¹³⁰ The frequency integration of the self-energy is based on the analytic continuation using either the 2-pole model¹³³ or Padé approximant as the analytic form.¹³⁴ For core levels, more sophisticated implementations are necessary.^{135,136} The standard implementation scales with $\mathcal{O}(N^4)$ with respect to system size N and enables the calculation of molecules up to a few hundred atoms on parallel supercomputers. Large molecules exceeding thousands of atoms can be treated by the low-scaling G_0W_0 implementation within CP2K,¹³⁷ which effectively scales with $\mathcal{O}(N^2)$ to $\mathcal{O}(N^3)$.

The GW equations should be in principle solved self-consistently. However, a fully self-consistent treatment is computationally very expensive.^{138,139} In CP2K, an approximate self-consistent scheme is available, where the wavefunctions ψ_n^{DFT} from DFT are employed and only the eigenvalues in G and W are iterated.^{128,130} That is, after completion of the G_0W_0 loop, the quasiparticle energies obtained from Eq. (38) are reinserted into the non-interacting Green's function G_0 and the screened Coulomb W_0 instead of the starting DFT eigenvalues. Via G_0 and W_0 , the change in the eigenvalues permeates to the self-energy and eventually to the energies ϵ_n .¹²⁰ This eigenvalue-self-consistent scheme (evGW) includes more physics than G_0W_0 but is still computationally tractable. Depending on the underlying DFT functional, evGW improves the agreement of the HOMO–LUMO gaps with experiment by 0.1–0.3 eV compared to G_0W_0 .^{128,130} For lower lying states, the improvement over G_0W_0 might be not as consistent.¹⁴⁰

Applications of the GW implementation in CP2K have been focused on graphene-based nanomaterials on gold surfaces

complementing scanning tunneling spectroscopy with GW calculations to validate the molecular geometry and to obtain information about the spin configuration.¹⁴¹ The excitation process generates an additional charge on the molecule. As a response, an image charge (IC) is formed inside the metallic surface, which causes occupied states to move up in energy and unoccupied states to move down. The HOMO–LUMO gap of the molecule is thus significantly lowered on the surface compared to the gas phase [see Fig. 2(c)]. This effect has been accounted for by an IC model,¹⁴² which is implemented in CP2K. Adding the IC correction on-top of a gas phase evGW calculation of the isolated molecule, CP2K can efficiently compute HOMO–LUMO gaps of physisorbed molecules on metal surfaces.¹⁴¹

However, if not explicitly stated otherwise, we will confine ourselves to conventional KS-DFT from now on.

V. CONSTRAINED DENSITY FUNCTIONAL THEORY

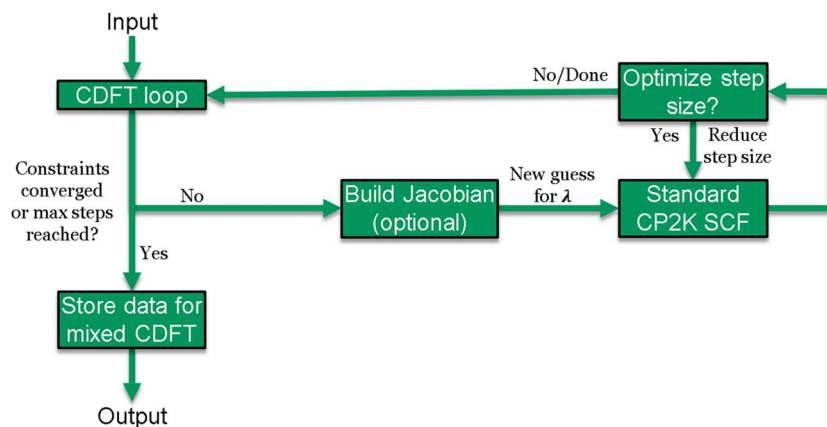
The CDFT method facilitates to construct charge and/or spin localized states, which are needed in a number of applications,¹⁴³ such as

- studying charge transfer (CT) phenomena and calculating electronic couplings (e.g., using the Marcus theory approach),
- correcting spurious charge delocalization due to self-interaction error, and
- parameterizing model Hamiltonians (e.g., the Heisenberg spin Hamiltonian).

The basic theory underlying CDFT has been derived by Wu and Van Voorhis,^{144,145} whereas the present implementation within CP2K is described in detail elsewhere,^{146,147} which is why both are only very briefly summarized here.

The charge and spin localized states are created by enforcing electron and spin density localization within atom centered regions of space. To this effect, the KS energy functional $E_{\text{KS}}[\rho]$ is augmented by additional constraint potentials, i.e.,

$$E_{\text{CDFT}}[\rho, \lambda] = \max_{\lambda} \min_{\rho} \left(E_{\text{KS}}[\rho] + \sum_c \lambda_c \left[\sum_{i=\uparrow, \downarrow} \int d\mathbf{r} w_c^i(\mathbf{r}) \rho^i(\mathbf{r}) - N_c \right] \right), \quad (39)$$



where $\lambda = [\lambda_1, \lambda_2, \dots]^T$ are the constraint Lagrangian multipliers, which can be thought of as the strengths of the constraint potentials, N_c is the target value of the corresponding constraint, whereas $w_c^i(\mathbf{r})$ is an atom centered weight function. The latter is constructed as a normalized sum over a set of selected constraint atoms \mathcal{C} ; hence,

$$w_c^i(\mathbf{r}) = \frac{\sum_{j \in \mathcal{C}} c_j P_j(\mathbf{r})}{\sum_{j \in \mathcal{N}} P_j(\mathbf{r})}, \quad (40)$$

where c_j are atomic coefficients that determine how each atom is included in the constraint, P_j is the so-called cell function that controls the volume occupied by atom j according to some population analysis method, whereas \mathcal{N} is the set of all atoms in a system. Different types of constraints can be constructed by modifying $w_c^i(\mathbf{r})$ according to the following conventions:

- charge constraint ($\rho^\uparrow + \rho^\downarrow$): $w^i = w^\uparrow = w^\downarrow$,
- magnetization constraint ($\rho^\uparrow - \rho^\downarrow$): $w^i = w^\uparrow = -w^\downarrow$, and
- spin specific constraint ($\rho^{\uparrow/\downarrow}$): $w^i = w^{\uparrow/\downarrow}$, $w^{\downarrow/\uparrow} = 0$.

In CP2K, the Becke and Hirshfeld space partitioning schemes are implemented as constraint weight functions. Using CDFT within AIMD or to optimize the geometry, however, the following additional force term arises due to the employed constraints:

$$\mathbf{F}_{I,c} = -\lambda_c \int d\mathbf{r} \frac{\partial w_c(\mathbf{r})}{\partial \mathbf{R}_I} \rho(\mathbf{r}). \quad (41)$$

As illustrated in Fig. 3, $E_{\text{CDFT}}[\rho, \lambda]$ is optimized self-consistently using a two-tiered approach: one external optimization loop for the constraints and an inner loop to converge the electronic structure, as described in detail in Sec. VIII. By definition, all constraints are satisfied when

$$\mathbf{c}(\lambda) = \left[\sum_{i=\uparrow, \downarrow} \int d\mathbf{r} w_c^i(\mathbf{r}) \rho^i(\mathbf{r}) - N_1, \dots \right]^T = \mathbf{0}. \quad (42)$$

The constraint Lagrangian multipliers λ can therefore be optimized by minimizing the constraint error expression $\max_{\lambda} |\mathbf{c}(\lambda)|$ using root-finding algorithms. For Newton and quasi-Newton optimizers, a new guess for λ at step n is generated according to the following iteration formula:

FIG. 3. Schematic of the CDFT SCF procedure. The constraint Lagrangians λ are first optimized in the outer CDFT loop, their values are subsequently fixed, and the electron density corresponding to these fixed values is solved like in traditional CP2K DFT simulations. The control is then returned to the outer CDFT loop, where the convergence of the constraints is checked. This iteration process is repeated until convergence is achieved.

$$\lambda_n = \lambda_{n-1} - \alpha J_n^{-1} c(\lambda_{n-1}), \quad (43)$$

where $\alpha \in (0, 1]$ is the step size and J^{-1} is the inverse Jacobian matrix, which is approximated by means of finite differences.

A. Mixed constrained density functional theory

Additional properties can be calculated from the interactions between CDFT states. Since in CP2K, these types of simulations, where multiple CDFT states are treated in parallel, leverage the mixed force methods described in Sec. XII, we will refer to them as mixed CDFT simulations, which are useful for a variety of different functions including

- calculating charge transfer kinetics parameters and
- performing configuration interaction (CI) calculations within the basis of CDFT states.

The theoretical concepts related to mixed CDFT calculations are best introduced through an example. Consider the following one electron transfer process $X^- + Y \rightarrow X + Y^-$. Denote the initial and final states of this reaction as A and B, respectively. Now, according to the Marcus theory of electron transfer, the charge transfer rate of this reaction is given by the rate equation

$$k_{ab} = \frac{2\pi}{\hbar} \frac{\langle |H_{ab}|^2 \rangle_T}{\sqrt{4\pi k_B T \xi}} \exp\left(-\frac{(\xi + \Delta A)^2}{4\pi k_B T \xi}\right), \quad (44)$$

where ΔA is the reaction free energy, ξ is the solvent reorganization energy, and $|H_{ab}|$ is the electronic coupling. The first two quantities can be obtained from free energy simulations, whereas the last is rigorously defined as the interaction energy between wavefunctions Ψ representing the two reaction states, i.e.,

$$H_{ab} = \langle \Psi_a | \mathcal{H} | \Psi_b \rangle, \quad (45)$$

where \mathcal{H} is the many-electron Hamilton operator. The usefulness of the electronic coupling quantity is not limited to the Marcus rate equation, but it is also a central quantity in other charge transfer theories, as well as in CDFT-based CI.¹⁴⁸

Since the true interacting many-electron wavefunctions or the Hamiltonian are not available within CDFT, the electronic coupling is instead approximated using the CDFT surrogates

$$H_{AB} \approx \langle \Phi_A | \mathcal{H}_{KS} | \Phi_B \rangle = E_B S_{AB} - \sum_c \lambda_c^B \mathbf{W}_c^{AB}, \quad (46a)$$

$$H_{BA} \approx \langle \Phi_B | \mathcal{H}_{KS} | \Phi_A \rangle = E_A S_{BA} - \sum_c \lambda_c^A \mathbf{W}_c^{BA}, \quad (46b)$$

where Φ are the CDFT KS determinants, E_I is the CDFT energy of state I , $S_{AB} = \langle \Phi_A | \Phi_B \rangle$, and \mathbf{W}_c^{AB} are the weight function matrices defined by

$$\mathbf{W}_c^{AB} = \langle \Phi_A | w_c^B(\mathbf{r}) | \Phi_B \rangle. \quad (47)$$

In the above expressions, capital subscripts have been used to emphasize the fact that the CDFT determinants are in general non-orthogonal. The electronic couplings and overlaps are collected into matrices \mathbf{H} and \mathbf{S} , respectively. The off-diagonal elements of \mathbf{H} are

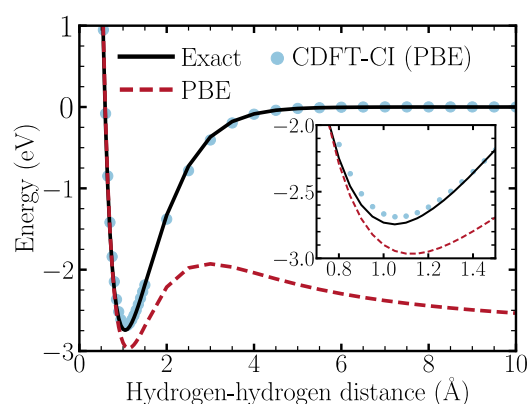


FIG. 4. Illustration of the DFT self-interaction error using the PBE XC functional for the reaction $\text{H}_2^+ \rightarrow \text{H}^+ + \text{H}$. However, employing the fragment constraint states $|\text{H}^+\text{H}\rangle$ and $|\text{HH}^+\rangle$ as the basis, the correct profile can be recovered with CDFT-CI.

not symmetric. The matrix is converted to the symmetric form by setting

$$\begin{aligned} H'_{AB} = H'_{BA} &= \frac{H_{AB} + H_{BA}}{2} \\ &= \frac{E_A + E_B}{2} S_{AB} - \sum_c \left\langle \Phi_A \left| \frac{\lambda_c^A w_c^A(\mathbf{r}) + \lambda_c^B w_c^B(\mathbf{r})}{2} \right| \Phi_B \right\rangle. \end{aligned} \quad (48)$$

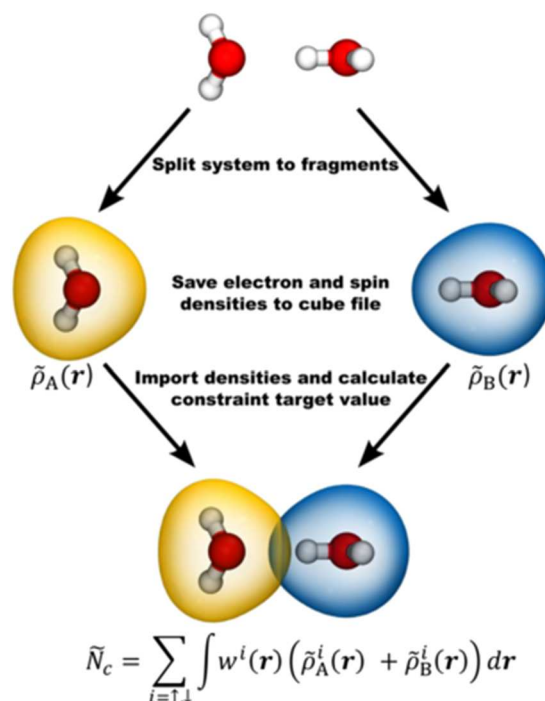


FIG. 5. Using a fragment-based CDFT constraint, the system is first divided into two fragments with atomic positions fixed in the same configuration as in the full system. The electron and spin densities of the fragment systems are then saved to cube files and subsequently used as input files for the CDFT calculation, where the constraint target value is calculated from the superimposed fragment densities.

The resulting matrix \mathbf{H}' is then orthogonalized to yield the final electronic coupling.

A pathological example of a system where the self-interaction error of common DFT XC functionals leads to unphysical results is the simple dissociation reaction $\text{H}_2^+ \rightarrow \text{H}^+ + \text{H}$. Even though this system contains only one electron, the dissociation profile obtained with the Perdew-Burke-Ernzerhof (PBE) XC functional notably deviates from the exact HF curve, as shown in Fig. 4. However, using CDFT states as the basis of a CI calculation, it is possible to correct for the self-interaction error of the H_2^+ ion. In fact, using the PBE XC functional, CDFT-CI is able to reproduce the exact dissociation profile. Very briefly, CDFT-CI simulations involve representing the system's wavefunction as a linear combination of multiple CDFT states, where the charge/spin density is constrained differently in different states. For that purpose, so-called fragment-based constraints are employed, where the constraint target value is calculated from the superposition of isolated fragment densities according to the scheme shown in Fig. 5. The CI expansion coefficients and energies are then obtained by solving a generalized eigenvalue equation, where the effective Hamilton matrix describes how the CDFT states interact with each other.

VI. DENSITY FUNCTIONAL PERTURBATION THEORY

Several experimental observables are measured by perturbing the system and then observing its response; hence, they can be obtained as derivatives of the energy or density with respect to some specific external perturbation. In the common perturbation approach, the perturbation is included in the Hamiltonian, i.e., as an external potential, then the electronic structure is obtained by applying the variational principle and the changes in the expectation values are evaluated. The perturbation Hamiltonian defines the specific problem. The perturbative approach applied in the framework of DFT turns out to perform better than the straightforward numerical methods, where the total energy is computed after actually perturbing the system.^{149,150} For all kinds of properties related to derivatives of the total energy, DFPT is derived from the following extended energy functional:

$$\mathcal{E}[\{\phi_i\}] = \mathcal{E}_{\text{KS}}[\{\phi_i\}] + \lambda \mathcal{E}_{\text{pert}}[\{\phi_i\}], \quad (49)$$

where the external perturbation is added in the form of a functional and λ is a small perturbative parameter representing the strength of the interaction with the static external field.^{151,152} The minimum of the functional is expanded perturbatively in powers of λ as

$$E = E^{(0)} + \lambda E^{(1)} + \lambda^2 E^{(2)} + \dots, \quad (50)$$

whereas the corresponding minimizing orbitals are

$$\phi_i = \phi_i^{(0)} + \lambda \phi_i^{(1)} + \lambda^2 \phi_i^{(2)} + \dots. \quad (51)$$

If the expansion of the wavefunction up to an order $(n-1)$ is known, then the variational principle for the $2n$ th-order derivative of the energy is given by

$$E^{(2n)} = \min_{\phi_i^{(n)}} \left\{ \mathcal{E} \left[\sum_{k=0}^n \lambda^k \phi_i^{(k)} \right] \right\} \quad (52)$$

under the constraint

$$\sum_{k=0}^n \langle \phi_i^{(n-k)} | \phi_j^{(k)} \rangle = 0. \quad (53)$$

For zero-order, the solution is obtained from the ground state KS equations. The second-order energy is variational in the first-order wavefunction and is obtained as

$$\begin{aligned} E^{(2)}[\{\phi_i^{(0)}\}, \{\phi_i^{(1)}\}] &= \sum_{ij} \langle \phi_i^{(1)} | \mathcal{H}^{(0)} \delta_{ij} - \Lambda_{ij} | \phi_j^{(1)} \rangle \\ &+ \sum_i \left[\langle \phi_i^{(1)} | \frac{\delta \mathcal{E}_{\text{pert}}}{\delta \langle \phi_i^{(0)} |} + \frac{\delta \mathcal{E}_{\text{pert}}}{\delta | \phi_i^{(0)} \rangle} | \phi_i^{(1)} \rangle \right] \\ &+ \frac{1}{2} \int d\mathbf{r} \int d\mathbf{r}' \mathcal{K}(\mathbf{r}, \mathbf{r}') n^{(1)}(\mathbf{r}) n^{(1)}(\mathbf{r}'). \end{aligned} \quad (54)$$

The electron density is also expanded in powers of λ , and the first-order term reads as

$$n^{(1)}(\mathbf{r}) = \sum_i f_i [\phi_i^{(0)*}(\mathbf{r}) \phi_i^{(1)}(\mathbf{r}) + \phi_i^{(1)*}(\mathbf{r}) \phi_i^{(0)}(\mathbf{r})]. \quad (55)$$

The Lagrange multipliers Λ_{ij} are the matrix elements of the zeroth-order Hamiltonian, which is the KS Hamiltonian. Hence,

$$\Lambda_{ij} = \langle \phi_i^{(0)} | \mathcal{H}^{(0)} | \phi_j^{(0)} \rangle, \quad (56)$$

and the second-order energy kernel is

$$\mathcal{K}(\mathbf{r}, \mathbf{r}') = \left. \frac{\delta^2 \mathcal{E}_{\text{Hxc}}[n]}{\delta n(\mathbf{r}) \delta n(\mathbf{r}')} \right|_{n^{(0)}}, \quad (57)$$

where \mathcal{E}_{Hxc} represents the sum of the Hartree and the XC energy functionals. Thus, the evaluation of the kernel requires the second-order functional derivative of the XC functionals.

The second-order energy is variational with respect to $\{\phi_i^{(1)}\}$, where the orthonormality condition of the total wavefunction gives at the first-order

$$\langle \phi_i^{(0)} | \phi_j^{(1)} \rangle + \langle \phi_i^{(1)} | \phi_j^{(0)} \rangle = 0, \quad \forall i, j. \quad (58)$$

This also implies the conservation of the total charge.

The perturbation functional can often be written as the expectation value of a perturbation Hamiltonian

$$\mathcal{H}^{(1)} = \frac{\delta \mathcal{E}_{\text{pert}}}{\delta \langle \phi_i^{(0)} |} + \int d\mathbf{r}' \mathcal{K}(\mathbf{r}, \mathbf{r}') n^{(1)}(\mathbf{r}'). \quad (59)$$

However, the formulation through an arbitrary functional also allows orbital specific perturbations. The stationary condition then yields the inhomogeneous, non-linear system for $\{\phi_i^{(1)}\}$, i.e.,

$$\begin{aligned} & - \sum_j (\mathcal{H}^{(0)} \delta_{ij} - \Lambda_{ij}) | \phi_j^{(1)} \rangle \\ & = \mathcal{P} \left(\int d\mathbf{r}' \mathcal{K}(\mathbf{r}, \mathbf{r}') n^{(1)}(\mathbf{r}') | \phi_i^{(0)} \rangle + \frac{\delta \mathcal{E}_{\text{pert}}}{\delta \langle \phi_i^{(0)} |} \right), \end{aligned} \quad (60)$$

where $\mathcal{P} = 1 - \sum_j |\phi_j^{(0)}\rangle\langle\phi_j^{(0)}|$ is the projector upon the unoccupied states. Note that the right-hand side still depends on the $\{\phi_i^{(1)}\}$ via the perturbation density $n^{(1)}$. In our implementation, the problem is solved directly using a preconditioned conjugate-gradient (CG) minimization algorithm.

A. Polarizability

One case where the perturbation cannot be expressed in a Hamiltonian form is the presence of an external electric field, which couples with the electric polarization $\mathbf{P}^{\text{el}} = e\langle\mathbf{r}\rangle$, where $i\langle\mathbf{r}\rangle$ is the expectation value of the position operator for the system of N electrons. In the case of periodic systems, the position operator is ill-defined, and we use the modern theory of polarization in the Γ -point-only to write the perturbation in terms of the Berry phase,

$$\gamma_\alpha = \text{Im} \log \det \mathbf{Q}^{(\alpha)}, \quad (61)$$

where the matrix is defined as

$$\mathbf{Q}_{ij}^{(\alpha)} = \langle\phi_i|\exp[i2\pi\mathbf{h}_\alpha^{-1}\cdot\mathbf{r}]\phi_j\rangle \quad (62)$$

and $\mathbf{h} = [\mathbf{a}, \mathbf{b}, \mathbf{c}]$ is the 3×3 matrix defining the simulation cell and $\mathbf{h}_\alpha = (a_\alpha, b_\alpha, c_\alpha)$.^{153–157} The electric dipole is then given by

$$P_\alpha^{\text{el}} = \frac{e}{2\pi} h_\alpha \gamma_\alpha. \quad (63)$$

Through the coupling to an external electric field \mathbf{E}^{ext} , this induces a perturbation of the type

$$\lambda \mathcal{E}_{\text{pert}}[\{\phi_i\}] = - \sum_\alpha E_\alpha^{\text{ext}} P_\alpha^{\text{el}}, \quad (64)$$

where the perturbative parameter is the field component E_α^{ext} . The functional derivative $\delta\mathcal{E}_{\text{pert}}/\delta\langle\phi_i^{(0)}|$ can be evaluated using the formula for the derivative of a matrix with respect to a generic variable,¹⁵² which gives the perturbative term as

$$\frac{\delta \log \det \mathbf{Q}^{(\alpha)}}{\delta\langle\phi_i^{(0)}|} = \sum_j \left(\mathbf{Q}^{(\alpha)}\right)_{ij}^{-1} \exp[i2\pi\mathbf{h}_\alpha^{-1}\cdot\mathbf{r}]\phi_j. \quad (65)$$

Once the first-order correction to the set of the KS orbitals has been calculated, the induced polarization is

$$\begin{aligned} \delta P_\alpha^{\text{el}} = & - \sum_\beta \frac{e}{2\pi} h_\beta \text{Im} \left[\sum_{ij} \left(\langle\phi_i^{\beta(1)}|\exp[i2\pi\mathbf{h}_\alpha^{-1}\cdot\mathbf{r}]\phi_j^{(0)}\rangle \right. \right. \\ & \left. \left. + \langle\phi_i^{(0)}|\exp[i2\pi\mathbf{h}_\alpha^{-1}\cdot\mathbf{r}]\phi_j^{\beta(1)}\rangle \right) \left(\mathbf{Q}^{(\alpha)}\right)_{ij}^{-1} \right] E_\beta, \end{aligned} \quad (66)$$

while the elements of the polarizability tensor are obtained as $\alpha_{\alpha\beta} = \partial P_\alpha^{\text{el}}/\partial E_\beta$.

The polarizability can be looked as the deformability of the electron cloud of the molecule by the electric field. In order for a molecular vibration to be Raman active, the vibration must be accompanied by a change in the polarizability. In the usual Placzek theory, ordinary Raman scattering intensities can be expressed in terms of the isotropic transition polarizability $\alpha^i = \frac{1}{3}\text{Tr}[\alpha]$ and the anisotropic transition polarizability $\alpha^a = \sum_{\alpha\beta} \frac{1}{2}(3\alpha_{\alpha\beta}\alpha_{\alpha\beta} - \alpha_{\alpha\alpha}\alpha_{\beta\beta})$.¹⁵⁸ The Raman scattering cross section can be related to the dynamical

autocorrelation function of the polarizability tensor. Along finite-temperature AIMD simulations, the polarizability can be calculated as a function of time.^{159,160} As the vibrational spectra are obtained by the temporal Fourier transformation of the velocity autocorrelation function, and the IR spectra from that of the dipole autocorrelation function,¹⁶¹ the depolarized Raman intensity can be calculated from the autocorrelation of the polarizability components.¹⁶²

B. Nuclear magnetic resonance and electron paramagnetic resonance spectroscopy

The development of the DFPT within the GAPW formalism allows for an all-electron description, which is important when the induced current density generated by an external static magnetic perturbation is calculated. The so induced current density determines at any nucleus A the nuclear magnetic resonance (NMR) chemical shift

$$\sigma_{\alpha\beta}^A = \frac{1}{c} \int \left[\frac{\mathbf{r} - \mathbf{R}_A}{|\mathbf{r} - \mathbf{R}_A|^3} \times \mathbf{j}_\alpha \right]_\beta d\mathbf{r} \quad (67)$$

and, for systems with net electronic spin 1/2, the electron paramagnetic resonance (EPR) \mathbf{g} -tensor

$$g_{\alpha\beta} = g_e \delta_{\alpha\beta} + \Delta g_{\alpha\beta}^{\text{ZKE}} + \Delta g_{\alpha\beta}^{\text{SO}} + \Delta g_{\alpha\beta}^{\text{SOO}}. \quad (68)$$

In the above expressions, \mathbf{R}_A is the position of the nucleus, \mathbf{j}_α is the current density induced by a constant external magnetic field applied along the α axis, and g_e is the free electron g -value. Among the different contributions to the \mathbf{g} -tensor, the current density dependent ones are the spin-orbit (SO) interaction,

$$\Delta g_{\alpha\beta}^{\text{SO}} = \frac{g_e - 1}{c} \int [\mathbf{j}_\alpha^\dagger(\mathbf{r}) \times \nabla V_{\text{eff}}^\dagger(\mathbf{r}) - \mathbf{j}_\alpha^\dagger(\mathbf{r}) \times \nabla V_{\text{eff}}^\dagger(\mathbf{r})]_\beta d\mathbf{r} \quad (69)$$

and the spin-other-orbit (SOO) interaction

$$\Delta g_{\alpha\beta}^{\text{SOO}} = 2 \int B_{\alpha\beta}^{\text{corr}}(\mathbf{r}) n^{\text{spin}}(\mathbf{r}) d\mathbf{r}, \quad (70)$$

where

$$B_{\alpha\beta}^{\text{corr}}(\mathbf{r}) = \frac{1}{c} \int \left[\frac{\mathbf{r}' - \mathbf{r}}{|\mathbf{r}' - \mathbf{r}|^3} \times (\mathbf{j}_\alpha(\mathbf{r}') - \mathbf{j}_\alpha^{\text{spin}}(\mathbf{r}')) \right]_\beta d\mathbf{r}', \quad (71)$$

which also depends on the spin density n^{spin} and the spin-current density \mathbf{j}^{spin} . Therein, V_{eff}^\dagger is an effective potential in which the spin up electrons are thought to move (similarly V_{eff}^\dagger for spin down electrons), whereas $B_{\alpha\beta}^{\text{corr}}$ is the β component of the magnetic field originating from the induced current density along α . The SO-coupling is the leading correction term in the computation of the g -tensor. It is relativistic in origin and therefore becomes much more important for heavy elements. In the current CP2K implementation, the SO term is obtained by integrating the induced spin-dependent current densities and the gradient of the effective potential over the simulation cell. The effective one-electron operator replaces the computationally demanding two-electrons integrals.¹⁶³ A detailed discussion on the impact of the various relativistic and SO approximations, which are implemented in the various codes, is provided by Deyne *et al.*¹⁶⁴

In the GAPW representation, the induced current density is decomposed with the same scheme applied for the electron density distinguishing among the soft contribution to the total current density, and the local hard and local soft contributions, i.e.,

$$\mathbf{j}(\mathbf{r}) = \tilde{\mathbf{j}}(\mathbf{r}) + \sum_A (\mathbf{j}_A(\mathbf{r}) + \tilde{\mathbf{j}}_A(\mathbf{r})). \quad (72)$$

In the linear response approach, the perturbation Hamiltonian at the first-order in the field strength is

$$\mathcal{H}^{(1)} = \frac{e}{m} \mathbf{p} \cdot \mathbf{A}(\mathbf{r}), \quad (73)$$

where \mathbf{p} is the momentum operator and \mathbf{A} is the vector potential representing the field \mathbf{B} . Thus,

$$\mathbf{A}(\mathbf{r}) = \frac{1}{2} (\mathbf{r} - \mathbf{d}(\mathbf{r})) \times \mathbf{B}, \quad (74)$$

with the cyclic variable $\mathbf{d}(\mathbf{r})$ being the gauge origin. The induced current density is calculated as the sum of orbital contributions \mathbf{j}_i and can be separated in a diamagnetic term $\mathbf{j}_i^d(\mathbf{r}) = \frac{e^2}{m} \mathbf{A}(\mathbf{r}) |\phi_i^{(0)}(\mathbf{r})|^2$ and a paramagnetic term $\mathbf{j}_i^p(\mathbf{r}) = \frac{e^2}{m} \langle \phi_i^{(0)} | [\mathbf{p}|\mathbf{r}\rangle\langle\mathbf{r}| + |\mathbf{r}\rangle\langle\mathbf{r}|\mathbf{p}] | \phi_i^{(1)} \rangle$. Both contributions individually are gauge dependent, whereas the total current is gauge-independent. The position operator appears in the definition of the perturbation operators and of the current density. In order to be able to deal with periodic systems, where the multiplicative position operator is not a valid operator, first we perform a unitary transformation of the ground state orbitals to obtain their maximally localized Wannier functions (MLWFs) counterpart.^{165–167} Hence, we use the alternative definition of the position operator, which is unique for each localized orbital and showing a sawtooth-shaped profile centered at the orbital's Wannier center.^{168–170}

Since we work with real ground state MOs, in the unperturbed state, the current density vanishes. Moreover, the first-order perturbation wavefunction is purely imaginary, which results in a vanishing first-order change in the electronic density $n^{(1)}$. This significantly simplifies the perturbation energy functional, since the second-order energy kernel can be skipped. The system of linear equations to determine the matrix of the expansion coefficients of the linear response orbitals $\mathbf{C}^{(1)}$ reads as

$$-i \sum_{\mu} \left(\mathcal{H}_{\nu\mu}^{(0)} \delta_{ij} - S_{\nu\mu} \langle \phi_i^{(0)} | \mathcal{H}^{(0)} | \phi_j^{(0)} \rangle \right) C_{\mu i} = \sum_{\mu} \mathcal{H}_{\nu\mu(j)}^{(1)} C_{\mu j}^{(0)}, \quad (75)$$

where i and j are the orbital indexes, ν and μ are the basis set function indexes, and $S_{\nu\mu}$ is an element of the overlap matrix. The optional subindex (j), labeling the matrix element of the perturbation operator, indicates that the perturbation might be orbital dependent. In our CP2K implementation,⁴³ the formalism proposed by Sebastiani *et al.* is employed,^{168,169,171} i.e., we split the perturbation in three types of operators, which are $\mathbf{L} = (\mathbf{r} - \mathbf{d}_j) \times \mathbf{p}$, the orbital angular momentum operator \mathbf{p} ; the momentum operator; and $\Delta_i = (\mathbf{d}_i - \mathbf{d}_j) \times \mathbf{p}$, the full correction operator. The vector \mathbf{d}_j is the Wannier center associated with the unperturbed j th orbital, thus making \mathbf{L} and Δ_i dependent on the unperturbed orbital to which they are applied. By using the Wannier center as a relative origin in the definition of \mathbf{L} , we introduce an individual reference system, which is then corrected by Δ_i . As a consequence, the response orbitals are given by nine sets

of expansion coefficients, as for each operator all three Cartesian components need to be individually considered. The evaluation of the orbital angular momentum contributions and of the momentum contributions can be done at the computational cost of just one total energy calculation. The full correction term, instead, requires one such calculation for each electronic state. This term does not vanish unless all \mathbf{d}_i are equal. However, in most circumstances, this correction is expected to be small, since it describes the reaction of state i to the perturbation of state j , which becomes negligible when the two states are far away. Once all contributions have been calculated, the x -component of the current density induced by the magnetic field along y is

$$j_{xy}(\mathbf{r}) = -\frac{1}{2c} \sum_{i\nu\mu} \left[C_{\nu i}^{(0)} \left(C_{\mu i}^{Ly} + (\mathbf{d}(\mathbf{r}) - \mathbf{d}_i)_x C_{\mu i}^{Pz} - (\mathbf{d}(\mathbf{r}) - \mathbf{d}_i)_z C_{\mu i}^{Px} - C_{\mu i}^{\Delta_{iy}} \right) \times (\nabla_x \varphi_\nu(\mathbf{r}) \varphi_\mu(\mathbf{r}) - \varphi_\nu(\mathbf{r}) \nabla_x \varphi_\mu(\mathbf{r})) \right] + (\mathbf{r} - \mathbf{d}(\mathbf{r}))_z n^{(0)}(\mathbf{r}), \quad (76)$$

where the first term is the paramagnetic contribution and the second term is the diamagnetic one.

The convergence of the magnetic properties with respect to the Gaussian basis set size is strongly dependent on the choice of the gauge. The available options in CP2K are the individual gauge for atoms in molecules (IGAIm) approach introduced by Keith and Bader,¹⁷² and the continuous set of gauge transformation (CSGT) approach.¹⁷³ The diamagnetic part of the current density vanishes identically when the CSGT approach is used, i.e., $\mathbf{d}(\mathbf{r}) = \mathbf{r}$. Yet, this advantage is weakened by the rich basis set required to obtain an accurate description of the current density close to the nuclei, which typically affects the accuracy within the NMR chemical shift. In the IGAIm approach, however, the gauge is taken at the closer nuclear center. Condensed-phase applications involve the calculation of NMR spectra of energy-storage materials,^{174,175} biomolecular^{176–178} and hydrogen-bonded systems,^{179–182} as well as the Overhauser dynamic nuclear polarization in solids.¹⁸³ Large-scale computations of NMR chemical shifts for extended paramagnetic solids were reported by Mondal *et al.*⁴⁴ They showed that the contact, pseudocontact, and orbital-shift contributions to paramagnetic NMR can be calculated by combining hyperfine couplings obtained by hybrid functionals with g -tensors and orbital shieldings computed using gradient-corrected functionals.¹⁸⁴

VII. TIME-DEPENDENT DENSITY FUNCTIONAL THEORY

The dynamics and properties of many-body systems in the presence of time-dependent potentials, such as electric or magnetic fields, can be investigated via TD-DFT.

A. Linear-response time-dependent density functional theory

Linear-response TD-DFT (LR-TDDFT)¹⁸⁵ is an inexpensive correlated method to compute vertical transition energies and oscillator strengths between the ground and singly excited electronic states. Optical properties are computed as a linear-response of the system to a perturbation caused by an applied weak electro-magnetic field.

The current implementation relies on Tamm-Dancoff and adiabatic approximations.¹⁸⁶ The Tamm-Dancoff approximation ignores electron de-excitation channels,^{187,188} thus reducing the LR-TDDFT equations to a standard Hermitian eigenproblem,¹⁸⁹ i.e.,

$$AX = \omega X, \quad (77)$$

where ω is the transition energy, X is a response eigenvector, and A is a response operator. In addition, the adiabatic approximation postulates the independence of the employed XC functional of time and leads to the following form for the matrix elements of the response operator:¹⁹⁰

$$A_{ia\sigma,jb\tau} = \delta_{ij}\delta_{ab}\delta_{\sigma\tau}(\epsilon_{a\sigma} - \epsilon_{i\sigma}) + (i_{\sigma}a_{\sigma}|j_{\tau}b_{\tau}) - c_{\text{HFX}}\delta_{\sigma\tau}(i_{\sigma}j_{\sigma}|a_{\tau}b_{\tau}) + (i_{\sigma}a_{\sigma}|f_{\text{xc};\sigma\tau}|j_{\tau}b_{\tau}). \quad (78)$$

In the above equation, the indices i and j stand for occupied spin-orbitals, whereas a and b indicated virtual spin-orbitals, and σ and τ refers to specific spin components. The terms $(i_{\sigma}a_{\sigma}|j_{\tau}b_{\tau})$ and $(i_{\sigma}a_{\sigma}|f_{\text{xc};\sigma\tau}|j_{\tau}b_{\tau})$ are standard electron repulsion and XC integrals over KS orbital functions $\{\phi(\mathbf{r})\}$ with corresponding KS orbital energies ϵ ; hence,

$$(i_{\sigma}a_{\sigma}|j_{\tau}b_{\tau}) = \int \varphi_{i\sigma}^*(\mathbf{r})\varphi_{a\sigma}(\mathbf{r}) \frac{1}{|\mathbf{r}-\mathbf{r}'|} \varphi_{j\tau}^*(\mathbf{r}')\varphi_{b\tau}(\mathbf{r}') d\mathbf{r}d\mathbf{r}' \quad (79a)$$

and

$$(i_{\sigma}a_{\sigma}|f_{\text{xc};\sigma\tau}|j_{\tau}b_{\tau}) = \int \varphi_{i\sigma}^*(\mathbf{r})\varphi_{a\sigma}(\mathbf{r}) f_{\text{xc};\sigma\tau}(\mathbf{r},\mathbf{r}') \times \varphi_{j\tau}^*(\mathbf{r}')\varphi_{b\tau}(\mathbf{r}') d\mathbf{r}d\mathbf{r}'. \quad (79b)$$

Here, the XC kernel $f_{\text{xc};\sigma\tau}(\mathbf{r},\mathbf{r}')$ is simply the second functional derivative of the XC functional E_{xc} over the ground state electron density $n^{(0)}(\mathbf{r})$,¹⁹¹ hence,

$$f_{\text{xc};\sigma\tau}(\mathbf{r},\mathbf{r}') = \frac{\delta^2 E_{\text{xc}}[n](\mathbf{r})}{\delta n_{\sigma}(\mathbf{r}')\delta n_{\tau}(\mathbf{r}')} \bigg|_{n=n^{(0)}}. \quad (80)$$

To solve Eq. (77), the current implementation uses a block Davidson iterative method.¹⁹² This scheme is flexible enough and allows us to tune the performance of the algorithm. In particular, it supports hybrid exchange functionals along with many acceleration techniques (see Sec. III), such as integral screening,⁵¹ truncated Coulomb operator,⁵² and ADMM.⁵³ Whereas in most cases the same XC functional is used to compute both the ground state electron density and the XC kernel, separate functionals are also supported. This can be used, for instance, to apply a long-term correction to the truncated Coulomb operator during the LR-TDDFT stage,⁵² when such correction has been omitted during the reference ground state calculation. The action of the response operator on the trial vector X for a number of excited states may also be computed simultaneously. This improves load-balancing and reduces communication costs, allowing a larger number of CPU cores to be effectively utilized.

1. Applications

The favorable scaling and performance of the LR-TDDFT code have been exploited to calculate the excitation energies of various systems with 1D, 2D, and 3D periodicities, such as cationic

defects in aluminosilicate imogolite nanotubes,¹⁹³ as well as surface and bulk canonical vacancy defects in MgO and HfO₂.¹⁸⁶ Throughout, the dependence of results on the fraction of Hartree-Fock exchange was explored and the accuracy of the ADMM approximation was verified. The performance was found to be comparable to ground state calculations for systems of ≈ 1000 atoms, which were shown to be sufficient to converge localized transitions from isolated defects, within these medium to wide bandgap materials.

B. Real-time time-dependent density functional theory and Ehrenfest dynamics

Alternative to perturbation based methods, real-time propagation-based TDDFT is also available in CP2K. The real-time TDDFT formalism allows us to investigate non-linear effects and can be used to gain direct insights into the dynamics of processes driven by the electron dynamics. For systems in which the coupling between electronic and nuclear motion is of importance, CP2K provides the option to propagate cores and electrons simultaneously using the Ehrenfest scheme. Both methods are implemented in a cubic- and linear scaling form. The cubic scaling implementation is based on MO coefficients (MO-RTP), whereas the linear scaling version acts on the density matrix (P-RTP).

While the derivation of the required equations for origin independent basis functions is rather straightforward, additional terms arise for atom centered basis sets.¹⁹⁴ The time-evolution of the MO coefficients in a non-orthonormal Gaussian basis reads as

$$\dot{a}_{\alpha}^j = \sum_{\beta\gamma} S_{\alpha\beta}^{-1} (i\mathbf{H}_{\beta\gamma} + \mathbf{B}_{\beta\gamma}) a_{\alpha}^j, \quad (81)$$

whereas the corresponding nuclear equations of motion is given by

$$M_I \ddot{\mathbf{R}}_I = -\frac{\partial U(\mathbf{R},t)}{\partial \mathbf{R}_I} + \sum_{j=1}^{N_e} \sum_{\alpha,\beta} a_{\alpha}^{j*} \left(\mathbf{D}_{\alpha\beta}^I - \frac{\partial H_{\alpha\beta}}{\partial \mathbf{R}_I} \right) a_{\beta}^j. \quad (82)$$

Therein, \mathbf{S} and \mathbf{H} are the overlap and KS matrices, whereas M_I and \mathbf{R}_I are the position and mass of ion I , and $U(\mathbf{R},t)$ is the potential energy of the ion-ion interaction. The terms involving these variables represent the basis set independent part of the equations of motion. The additional terms containing matrices \mathbf{B} and \mathbf{D} are arising as a consequence of the origin dependence and are defined as follows:

$$\mathbf{D}^I = \mathbf{B}^{I+} (\mathbf{S}^{-1}\mathbf{H} - i\mathbf{S}^{-1}\mathbf{B}) + i\mathbf{C}^{I+} + (\mathbf{H}\mathbf{S}^{-1} + i\mathbf{B}^+\mathbf{S}^{-1})\mathbf{B}^I + i\mathbf{C}^I, \quad (83)$$

with

$$\begin{aligned} \mathbf{B}_{\alpha\beta} &= \left\langle \phi_{\alpha} \left| \frac{d}{dt} \phi_{\beta} \right. \right\rangle, & \mathbf{B}_{\alpha\beta}^+ &= \left\langle \frac{d}{dt} \phi_{\alpha} \left| \phi_{\beta} \right. \right\rangle, \\ \mathbf{B}^I &= \left\langle \phi_{\alpha} \left| \frac{d}{d\mathbf{R}^I} \phi_{\beta} \right. \right\rangle, & \mathbf{B}^{I+} &= \left\langle \frac{d}{d\mathbf{R}^I} \phi_{\alpha} \left| \phi_{\beta} \right. \right\rangle, \\ \mathbf{C}^I &= \left\langle \frac{d}{dt} \phi_{\alpha} \left| \frac{d}{d\mathbf{R}^I} \phi_{\beta} \right. \right\rangle, & \mathbf{C}^{I+} &= \left\langle \frac{d}{d\mathbf{R}^I} \phi_{\alpha} \left| \frac{d}{dt} \phi_{\beta} \right. \right\rangle. \end{aligned} \quad (84)$$

For simulations with fixed ionic positions, i.e., when only the electronic wavefunction is propagated, the \mathbf{B} and \mathbf{D} terms are vanishing and Ref. 81 simplifies to just the KS term. The two most important propagators for the electronic wavefunction in CP2K are the exponential midpoint and the enforced time reversal symmetry (ETRS) propagators. Both propagators are based on matrix exponentials.

The explicit computation of it, however, can be easily avoided for both MO-RTP and P-RTP techniques. Within the MO-RTP scheme, the construction of a Krylov subspace $K_n(X, MO)$ together with Arnoldi's method allows for the direct computation of the action of the propagator on the MOs at a computational complexity of $O(N^2M)$. For the P-RTP approach, the exponential is applied from both sides, which allows the expansion into a series similar to the Baker–Campbell–Hausdorff expansion. This expansion is rapidly converging and in theory only requires sparse matrix–matrix multiplications. Unfortunately, pure linear scaling Ehrenfest dynamics seems to be impossible due to the non-exponential decay of the density matrix during such simulations.¹⁹⁵ This leads to a densification of the involved matrices and eventually cubic scaling with system size. However, the linear scaling version can be coupled with subsystem DFT to achieve true linear scaling behavior.

In subsystem DFT, as implemented in CP2K, follows the approach of Gordon and Kim.^{196,197} In this approach, the different subsystems are minimized independently with the XC functional of choice. The coupling between the different subsystem is added via a correction term using a kinetic energy functional,

$$E_{\text{corr}} = E_s[\rho] - \sum_i^{n_{\text{sub}}} E_s[\rho_i], \quad (85)$$

where ρ is the electron density of the full system and ρ_i is the electron density of subsystem i . Using an orbital-free density functional to compute the interaction energy does not affect the structure of the overlap, which is block diagonal in this approach. If P-RTP is applied to the Kim–Gordon density matrix, the block diagonal structure is preserved and linear scaling with the number of subsystems is achieved.

VIII. DIAGONALIZATION-BASED AND LOW-SCALING EIGENSOLVER

After the initial Hamiltonian matrix for the selected method has been built by CP2K, such as the KS matrix K in the case of a DFT-based method using the QUICKSTEP module, the calculation of the total (ground state) energy for the given atomic configuration is the next task. This requires an iterative self-consistent field (SCF) procedure as the Hamiltonian depends usually on the electronic density. In each SCF step, the eigenvalues and at least the eigenvectors of the occupied MOs have to be calculated. Various eigensolver schemes are provided by CP2K for that task:

- Traditional Diagonalization (TD),
- Pseudodiagonalization (PD),¹⁹⁸
- Orbital Transformation (OT) method,¹⁹⁹ and
- purification methods.²⁰⁰

The latter method, OT, is the method of choice concerning computational efficiency and scalability for growing system sizes. However, OT requires fixed integer MO occupations. Therefore, it is not applicable for systems with a very small or no bandgap, such as metallic systems, which need fractional (“smeared”) MO occupations. For very large systems in which even the scaling of the OT method becomes unfavorable, one has to resort to linear-scaling methods (see Sec. VIII E). By contrast, TD can also be applied with fractional MO occupations, but its cubic-scaling (N^3) limits the accessible

system sizes. In that respect, PD may provide significant speed-ups (factor of 2 or more) once a pre-converged solution has been obtained with TD.

In the following, we will restrict the description of the implemented eigensolver methods to spin-restricted systems, since the generalization to spin-unrestricted, i.e., spin-polarized systems is straightforward and CP2K can deal with both types of systems using each of these methods.

A. Traditional diagonalization

The TD scheme in CP2K employs an eigensolver either from the parallel program library ScaLAPACK (Scalable Linear Algebra PACKage)²⁰¹ or from the ELPA (Eigenvalue solvers for Petascale Applications)^{202,203} library to solve the general eigenvalue problem

$$K C = S C \epsilon, \quad (86)$$

where K is the KS and S is the overlap matrix. The eigenvectors C represent the MO coefficients, and ϵ are the corresponding MO eigenvalues. Unlike to PW codes, the overlap matrix S is not the unit matrix, since CP2K/QUICKSTEP employs atom-centered basis sets of non-orthogonal Gaussian-type functions (see Sec. II C). Thus, the eigenvalue problem is transformed to its special form

$$K C = U^T U C \epsilon, \quad (87a)$$

$$(U^T)^{-1} K U^{-1} C' = C' \epsilon, \quad (87b)$$

$$K' C' = C' \epsilon \quad (87c)$$

using a Cholesky decomposition of the overlap matrix

$$S = U^T U \quad (88)$$

as the default method for that purpose. Now, (87c) can simply be solved by a diagonalization of K' . The MO coefficient matrix C in the non-orthogonal basis is finally obtained by the back-transformation,

$$C' = U C. \quad (89)$$

Alternatively, a symmetric Löwdin orthogonalization instead of a Cholesky decomposition can be applied,²⁰⁴ i.e.,

$$U = S^{1/2}. \quad (90)$$

On the one hand, the calculation of $S^{1/2}$ as implemented in CP2K involves, however, a diagonalization of S , which is computationally more expensive than a Cholesky decomposition. On the other hand, however, linear dependencies in the basis set introduced by small Gaussian function exponents can be detected and eliminated when S is diagonalized. Eigenvalues of S smaller than 10^{-5} usually indicate significant linear dependencies in the basis set and filtering of the corresponding eigenvectors can help ameliorate numerical difficulties during the SCF iteration procedure. For small systems, the choice of the orthogonalization has no crucial impact on the performance since it has to be performed only once for each configuration during the initialization of the SCF run.

Only the occupied MOs are required for the build-up of the density matrix \mathbf{P} in the AO basis

$$\mathbf{P} = 2 \mathbf{C}_{\text{occ}} \mathbf{C}_{\text{occ}}^{\text{T}}. \quad (91)$$

This saves not only memory but also computational time since the orthonormalization of the eigenvectors is a time-consuming step. Usually, only 10%–20% of the orbitals are occupied when standard Gaussian basis sets are employed with CP2K.

The TD scheme is combined with methods to improve the convergence of the SCF iteration procedure. The most efficient SCF convergence acceleration is achieved by the direct inversion in the iterative sub-space (DIIS),^{205,206} exploiting the commutator relation

$$\mathbf{e} = \mathbf{K} \mathbf{P} \mathbf{S} - \mathbf{S} \mathbf{P} \mathbf{K} \quad (92)$$

between the KS and the density matrix, where the error matrix \mathbf{e} is zero for the converged density. The DIIS method can be very efficient in the number of iterations required to reach convergence starting from a sufficiently pre-converged density, which is significant if the cost of constructing the Hamiltonian matrix is larger than the cost of diagonalization.

B. Pseudodiagonalization

Alternative to TD, a pseudodiagonalization can be applied as soon as a sufficiently pre-converged wavefunction is obtained.¹⁹⁸ The KS matrix \mathbf{K}^{AO} in the AO basis is transformed into the MO basis in each SCF step via

$$\mathbf{K}^{\text{MO}} = \mathbf{C}^{\text{T}} \mathbf{K}^{\text{AO}} \mathbf{C} \quad (93)$$

using the MO coefficients \mathbf{C} from the preceding SCF step. The converged \mathbf{K}^{MO} matrix using TD is a diagonal matrix, and the eigenvalues are its diagonal elements. Already after a few SCF iteration steps, the \mathbf{K}^{MO} matrix becomes diagonally dominant. Moreover, the \mathbf{K}^{MO} matrix shows the following natural blocking

$$\begin{pmatrix} oo & ou \\ uo & uu \end{pmatrix} \quad (94)$$

due to the two MO sub-sets of \mathbf{C} , namely, the occupied (o) and the unoccupied (u) MOs. The eigenvectors are used during the SCF iteration to calculate the new density matrix [see Eq. (91)], whereas the eigenvalues are not needed. The total energy only depends on the occupied MOs, and thus, a block diagonalization, which decouples the occupied and unoccupied MOs, allows us to converge the wavefunctions. As a consequence, only all elements of the block ou or uo have to become zero, since \mathbf{K}^{MO} is a symmetric matrix. Hence, the transformation into the MO basis

$$\mathbf{K}_{\text{ou}}^{\text{MO}} = \mathbf{C}_o^{\text{T}} \mathbf{K}^{\text{AO}} \mathbf{C}_u \quad (95)$$

only has to be performed for that matrix block. Then, the decoupling can be achieved iteratively by consecutive 2×2 Jacobi rotations, i.e.,

$$\theta = \frac{\epsilon_q - \epsilon_p}{2 K_{pq}^{\text{MO}}}, \quad (96a)$$

$$t = \frac{\text{sgn}(\theta)}{|\theta| + \sqrt{1 + \theta^2}}, \quad (96b)$$

$$c = \frac{1}{\sqrt{t^2 + 1}}, \quad (96c)$$

$$s = tc, \quad (96d)$$

$$\tilde{\mathbf{C}}_p = c \mathbf{C}_p - s \mathbf{C}_q, \quad (96e)$$

$$\tilde{\mathbf{C}}_q = s \mathbf{C}_p + c \mathbf{C}_q, \quad (96f)$$

where the angle of rotation θ is determined by the difference of the eigenvalues of the MOs p and q divided by the corresponding matrix element K_{pq}^{MO} in the ou or uo block. The Jacobi rotations are computationally cheap as they can be performed with BLAS level 1 routines (DSCAL and DAXPY). The oo block is significantly smaller than the uu block, since only 10%–20% of the MOs are occupied using a standard basis set. Consequently, the ou or uo block also includes only 10%–20% of the \mathbf{K}^{MO} matrix. Furthermore, an expensive re-orthonormalization of the MO eigenvectors is not needed, since the Jacobi rotations preserve the orthonormality of the MO eigenvectors. Moreover, the number of non-zero blocks decreases rapidly when convergence is approached, which results in a decrease in the compute time for the PD part.

C. Orbital transformations

An alternative to the just described diagonalization-based algorithms is techniques that rely on direct minimization of the electronic energy functional.^{199,207–212} Convergence of this approach can in principle be guaranteed if the energy can be reduced in each step. The direct minimization approach is thus more robust. It also replaces the diagonalization step by having fewer matrix-matrix multiplications, significantly reducing the time-to-solution. This is of great importance for many practical problems, in particular, large systems that are difficult or sometimes even impossible to tackle with DIIS-like methods. However, preconditioners are often used in conjunction with direct energy minimization algorithms to facilitate faster and smoother convergence.

The calculation of the total energy within electronic structure theory can be formulated variationally in terms of an energy functional of the occupied single-particle orbitals that are constrained with respect to an orthogonality condition. With M orbitals, $\mathbf{C} \in \mathbb{R}^{N \times M}$ is given in a nonorthogonal basis consisting of N basis functions $\{\phi_i\}_{i=1}^N$ and its associated $N \times N$ overlap matrix \mathbf{S} , with element $S_{ij} = \langle \phi_i | \phi_j \rangle$. The corresponding constrained minimization problem is expressed as

$$\mathbf{C}^* = \arg \min_{\mathbf{C}} \{E[\mathbf{C}] \mid \mathbf{C}^{\text{T}} \mathbf{S} \mathbf{C} = \mathbf{1}\}, \quad (97)$$

where $E[\mathbf{C}]$ is an energy functional, \mathbf{C}^* is the minimizer of $E[\mathbf{C}]$ that fulfills the condition of orthogonality $\mathbf{C}^{\text{T}} \mathbf{S} \mathbf{C} = \mathbf{1}$, whereas $\arg \min$ stands for the argument of the minimum. The ground state energy is finally obtained as $E[\mathbf{C}^*]$. The form of the energy functional $E[\mathbf{C}]$ is

determined by the particular electronic structure theory used; in the case of hybrid Hartree–Fock/DFT, the equation reads as

$$E_{\text{HF/DFT}}[\mathbf{C}] = \text{Tr}[\mathbf{P}\mathbf{h}] + \frac{1}{2} \text{Tr}[\mathbf{P}(\mathbf{J}[\mathbf{P}] + \alpha_{\text{HF}}\mathbf{K}[\mathbf{P}])] + E_{\text{XC}}[\mathbf{P}], \quad (98)$$

where $\mathbf{P} = \mathbf{C}\mathbf{C}^T$ is the density matrix, whereas \mathbf{h} , \mathbf{J} , and \mathbf{K} are the core Hamiltonian, the Coulomb, and Hartree–Fock exact exchange matrices, respectively, and $E_{\text{XC}}[\mathbf{P}]$ is the XC energy.

Enforcing the orthogonality constraints within an efficient scheme poses a major hurdle in the direct minimization of $E[\mathbf{C}]$. Hence, in the following, we describe two different approaches implemented in CP2K.^{199,212}

1. Orthogonality constraints

a. Orbital transformation functions: OT/diag and OT/Taylor. To impose the orthogonality constraints on the orbitals, VandeVondele and Hutter reformulated the non-linear constraint on \mathbf{C} [see Eq. (97)] by a linear constraint on the auxiliary variable \mathbf{X} via

$$\mathbf{X}^* = \arg \min_{\mathbf{X}} \{E[\mathbf{C}(\mathbf{X})] \mid \mathbf{X}^T \mathbf{S} \mathbf{C}_0 = \mathbf{0}\} \quad (99)$$

and

$$\mathbf{C}^* = \mathbf{C}(\mathbf{X}^*), \quad (100)$$

where $\mathbf{X} \in \mathbb{R}^{N \times M}$ and \mathbf{C}_0 is a set of initial orbitals that fulfill the orthogonality constraints $\mathbf{C}_0^T \mathbf{S} \mathbf{C}_0 = \mathbf{1}$.¹⁹⁹ The OT is parametrized as follows:

$$\mathbf{C}(\mathbf{X}) = \mathbf{C}_0 \cos \mathbf{U} + \mathbf{X} \mathbf{U}^{-1} \sin \mathbf{U}, \quad (101)$$

where $\mathbf{U} = \mathbf{X}^T \mathbf{S} \mathbf{X}^{1/2}$. This parametrization ensures that $\mathbf{C}^T \mathbf{X} \mathbf{S} \mathbf{C} \mathbf{X} = \mathbf{1}$, for all \mathbf{X} satisfying the constraints $\mathbf{X}^T \mathbf{S} \mathbf{C}_0 = \mathbf{0}$. The matrix functions $\cos \mathbf{U}$ and $\mathbf{U}^{-1} \sin \mathbf{U}$ are evaluated either directly by diagonalization or by a truncated Taylor expansion in $\mathbf{X}^T \mathbf{S} \mathbf{X}$.²¹³

b. Orbital transformation based on a refinement expansion: OT/IR. In this method, Weber *et al.* replaced the constrained functional by an equivalent unconstrained functional $[\mathbf{C} \rightarrow f(\mathbf{Z})]$.²¹² The transformed minimization problem in Eq. (97) is then given by

$$\mathbf{Z}^* = \arg \min_{\mathbf{Z}} E[f(\mathbf{Z})] \quad (102)$$

and

$$\mathbf{C}^* = f(\mathbf{Z}^*), \quad (103)$$

where $\mathbf{Z} \in \mathbb{R}^{N \times M}$. The constraints have been mapped onto the matrix function $f(\mathbf{Z})$, which fulfills the orthogonality constraint $f^T(\mathbf{Z}) \mathbf{S} f(\mathbf{Z}) = \mathbf{1}$ for all matrices \mathbf{Z} . The main idea of this approach is to approximate the OT in Eq. (102) by $f_n(\mathbf{Z}) \approx f(\mathbf{Z})$, where $f_n(\mathbf{Z})$ is an approximate constraint function, which is correct up to some order $n + 1$ in $\delta \mathbf{Z} = \mathbf{Z} - \mathbf{Z}_0$, where $\mathbf{Z}_0^T \mathbf{S} \mathbf{Z}_0 = \mathbf{1}$. The functions derived by Niklasson for the iterative refinement of an approximate inverse matrix factorization are used to approximate $f(\mathbf{Z})$.²¹⁴ The first few refinement functions are given by

$$f_1(\mathbf{Z}) = \frac{1}{2} \mathbf{Z}(3 - \mathbf{Y}), \quad (104a)$$

$$f_2(\mathbf{Z}) = \frac{1}{8} \mathbf{Z}(15 - 10\mathbf{Y} + 3\mathbf{Y}^2), \quad (104b)$$

$$f_3(\mathbf{Z}) = \frac{1}{16} \mathbf{Z}(35 - 35\mathbf{Y} + 21\mathbf{Y}^2 - 5\mathbf{Y}^3), \quad (104c)$$

$$\dots \quad (104d)$$

where $\mathbf{Y} = \mathbf{Z}^T \mathbf{S} \mathbf{Z}$ and $\mathbf{Z} = \mathbf{Z}_0 + \delta \mathbf{Z}$. It can be shown that

$$f_n^T(\mathbf{Z}) \mathbf{S} f_n(\mathbf{Z}) - \mathbf{1} = \mathcal{O}(\delta \mathbf{Z}^{n+1}). \quad (105)$$

Using this general ansatz for $f_n(\mathbf{Z})$, it is possible to extend the accuracy to any finite order recursively by an iterative refinement expansion $f_n(\dots f_n(\mathbf{Z}) \dots)$.

2. Minimizer

a. Direct inversion of the iterative subspace. The DIIS method introduced by Pulay is an extrapolation technique based on minimizing the norm of a linear combination of gradient vectors.²⁰⁶ The problem is given by

$$c^* = \arg \min_c \left\{ \left\| \sum_{i=1}^m c_i \mathbf{g}_i \right\| \mid \sum_{i=1}^m c_i = 1 \right\}, \quad (106)$$

where \mathbf{g}_i is an error vector and c^* are the optimal coefficients.

The next orbital guess \mathbf{x}_{m+1} is obtained by the linear combination of the previous points using the optimal coefficients c^* , i.e.,

$$\mathbf{x}_{m+1} = \sum_{i=1}^m c_i^* (\mathbf{x}_i - \tau \mathbf{f}_i), \quad (107)$$

where τ is an arbitrary step size chosen for the DIIS method. The method simplifies to a steepest descent (SD) for the initial step $m = 1$. While the DIIS method converges very fast in most of the cases, it is not particularly robust. In CP2K, the DIIS method is modified to switch to SD when a DIIS step brings the solution toward an ascent direction. This safeguard makes DIIS more robust and is possible because the gradient of the energy functional is available.

b. Non-linear conjugate gradient minimization. Non-linear CG leads to a robust, efficient, and numerically stable energy minimization scheme. In non-linear CG, the residual is set to the negation of the gradient $\mathbf{r}_i = -\mathbf{g}_i$ and the search direction is computed by Gram–Schmidt conjugation of the residuals, i.e.,

$$\mathbf{d}_{i+1} = \mathbf{r}_{i+1} + \beta_{i+1} \mathbf{d}_i. \quad (108)$$

Several choices for updating β_{i+1} are available, and the Polak–Ribière variant with restart is used in CP2K.^{215,216}

$$\beta_{i+1} = \max \left(\frac{\mathbf{r}_{i+1}^T (\mathbf{r}_{i+1} - \mathbf{r}_i)}{\mathbf{r}_i^T \mathbf{r}_i}, 0 \right). \quad (109)$$

Similar to linear CG, a step length α_i is found that minimizes the energy function $f(\mathbf{x}_i + \alpha_i \mathbf{d}_i)$ using an approximate line search. The updated position becomes $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{d}_i$. In QUICKSTEP, a few

different line searches are implemented. The most robust is the golden section line search,²¹⁷ but the default quadratic interpolation along the search direction suffices in most cases. Regarding time-to-solution, the minimization performed with the latter quadratic interpolation is in general significantly faster than the golden section line search.

Non-linear CGs are generally preconditioned by choosing an appropriate preconditioner M that approximates f'' (see Sec. VIII C 3).

c. Quasi-Newton method. Newton's method can also be used to minimize the energy functional. The method is scale invariant, and the zigzag behavior that can be seen in the SD method is not present. The iteration for Newton's method is given by

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \beta \mathbf{H}(\mathbf{x}_k)^{-1} \mathbf{f}_k, \quad (110)$$

where \mathbf{H} is the Hessian matrix. On the one hand, the method exhibits super-linear convergence when the initial guess is close to the solution and $\beta = 1$, but, on the other hand, when the initial guess is further away from the solution, Newton's method may diverge. This divergent behavior can be suppressed by the introduction of line search or backtracking. As they require the inverse Hessian of the energy functional, the full Newton's method is in general too time-consuming or difficult to use. Quasi-Newton methods,²¹⁸ however, are advantageous alternatives to Newton's method when the Hessian is unavailable or is too expensive to compute. In those methods, an approximate Hessian is updated by analyzing successive gradient vectors. A quasi-Newton step is determined by

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \beta \mathbf{G}_k \mathbf{f}_k, \quad (111)$$

where \mathbf{G}_k is the approximate inverse Hessian at step k . Different update formulas exist to compute \mathbf{G}_k .²¹⁹ In QUICKSTEP, the Broyden's type 2 update is implemented to construct the approximate inverse Hessian with an adaptive scheme for estimating the curvature of the energy functional to increase the performance.²²⁰

3. Preconditioners

a. Preconditioning the non-linear minimization. Gradient based OT methods are guaranteed to converge but will exhibit slow convergence behavior if not appropriately preconditioned. A good reference for the optimization can be constructed from the generalized eigenvalue problem under the orthogonality constraint of Eq. (97) and its approximate second derivative

$$\frac{\partial^2 E}{\partial X_{\alpha i} \partial X_{\beta j}} = 2H_{\alpha\beta} \delta_{ij} - 2S_{\alpha\beta} \delta_{ij} \epsilon_i^0. \quad (112)$$

Therefore, close to convergence, the best preconditioners would be of the form

$$(\mathbf{H} - \mathbf{S} \epsilon_i^0)_{\alpha\beta}^{-1} \left(\frac{\delta E}{\delta X_{\beta i}} \right). \quad (113)$$

As this form is impractical, requiring a different preconditioner for each orbital, a single positive definite preconditioning matrix \mathbf{P} is constructed approximating

$$\mathbf{P}(\mathbf{H} - \mathbf{S} \epsilon) \mathbf{x} - \mathbf{x} \approx 0. \quad (114)$$

In CP2K, the closest approximation to this form is the FULL_ALL preconditioner. It performs an orbital dependent eigenvalue shift of \mathbf{H} . In this way, positive definiteness is ensured with minimal modifications. The downside is that the eigenvalues of \mathbf{H} have to be computed at least once using diagonalization and thus scales as $\mathcal{O}(N^3)$.

To overcome this bottleneck, several more approximate though lower scaling preconditioners have been implemented within CP2K. The simplest assume $\epsilon = \mathbf{1}$ and $\mathbf{H} = \mathbf{T}$, with \mathbf{T} being the kinetic energy matrix (FULL_KINETIC) or even $\mathbf{H} = \mathbf{0}$ (FULL_S_INVERSE) as viable approximations. These preconditioners are obviously less sophisticated. However, they are linear-scaling in their construction as \mathbf{S} and \mathbf{T} are sparse and still lead to accelerated convergence. Hence, these preconditioners are suitable for large-scale simulations.

For many systems, the best trade-off between the quality and the cost of the preconditioner is obtained with the FULL_SINGLE_INVERSE preconditioner. Instead of shifting all orbitals separately, only the occupied eigenvalues are inverted. Thus, making the orbitals closest to the bandgap most active in the optimization. The inversion of the spectrum can be done without the need for diagonalization by

$$\mathbf{P} = \mathbf{H} - 2\mathbf{S} \mathbf{C}_0 (\mathbf{C}_0^T \mathbf{H} \mathbf{C}_0 + \delta) \mathbf{C}_0^T \mathbf{S} - \epsilon \mathbf{S}, \quad (115)$$

where δ represents an additional shift depending on the HOMO energy, which ensures positive definiteness of the preconditioner matrix. It is important to note that the construction of this preconditioner matrix can be done with a complexity of $\mathcal{O}(NM^2)$ in the dense case and is therefore of the same complexity as the rest of the OT algorithm.

b. Reduced scaling and approximate preconditioning. All of the abovementioned preconditioners still require the inversion of the preconditioning matrix \mathbf{P} . In dense matrix algebra, this leads to an $\mathcal{O}(N^3)$ scaling behavior independent of the chosen preconditioner. For large systems, the inversion of \mathbf{P} will become the dominant part of the calculation when low-scaling preconditioners are used. As Schiffmann and VandeVondele had shown,²²¹ sparse matrix techniques are applicable for the inversion of the low-scaling preconditioners and can be activated using the INVERSE_UPDATE option as the preconditioner solver. By construction, the preconditioner matrix is symmetric and positive definite. This allows for the use of Hotelling's iterations to compute the inverse of \mathbf{P} as

$$\mathbf{P}_{i+1}^{-1} = \alpha \mathbf{P}_i^{-1} (2\mathbf{I} - \mathbf{P} \alpha \mathbf{P}_i^{-1}). \quad (116)$$

Generally, the resulting approximants to the inverse become denser and denser.²²² In CP2K, this is dealt with aggressive filtering on \mathbf{P}_{i+1}^{-1} . Unfortunately, there are limits to the filtering threshold. While the efficiency of the preconditioner is not significantly affected by the loss of accuracy (see Sec. XV B), the Hotelling iterations may eventually become unstable.²²³

Using a special way to determine the initial alpha based on the extremal eigenvalues of $\mathbf{P} \mathbf{P}_0^{-1}$, it can be shown that any positive definite matrix can be used as the initial guess for the Hotelling iterations.²²⁴ For AIMD simulations or geometry optimizations, this means the previous inverse can be used as the initial guess as the changes in \mathbf{P} are expected to be small.²²⁵ Therefore, only very few iterations are required until convergence after the initial approximation for the inverse is obtained.

D. Purification methods

Linear-scaling DFT calculations can also be achieved by purifying the KS matrix \mathbf{K} directly into the density matrix \mathbf{P} without using the orbitals \mathbf{C} explicitly.²⁰⁰ These density matrix-based methods exploit the fact that the KS matrix \mathbf{K} and the density matrix \mathbf{P} have by definition the same eigenvectors \mathbf{C} and that a purification maps eigenvalues ϵ_i of \mathbf{K} to eigenvalues f_i of \mathbf{P} via the Fermi–Dirac function

$$f_i = \frac{1}{\exp\left(\frac{\epsilon_i - \mu}{k_B T}\right) + 1}, \quad (117)$$

with the chemical potential μ , the Boltzmann constant k_B , and electron temperature T . In practice, the purification is computed by an iterative procedure that is constructed to yield a linear-scaling method for sparse KS matrices. By construction, such purifications are usually grand-canonical purifications so that additional measures such as modifications to the algorithms or additional iterations to find the proper value of the chemical potential are necessary to allow for canonical ensembles. In CP2K, the trace-resetting fourth-order method (TRS4²²⁶), the trace-conserving second order method (TC2²²⁷), and the purification via the sign function (SIGN¹¹⁵) are available. Additionally, CP2K implements an interface to the PEXSI (Pole EXpansion and Selected Inversion) library,^{228–231} which allows us to evaluate selected elements of the density matrix as the Fermi–Dirac function of the KS matrix via a pole expansion.

E. Sign-method

The sign function of a matrix

$$\text{sign}(\mathbf{A}) = \mathbf{A}(\mathbf{A}^2)^{-1/2} \quad (118)$$

can be used as a starting point for the construction of various linear-scaling algorithms.²³² The relation

$$\text{sign}\begin{pmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{I} & \mathbf{0} \end{pmatrix} = \text{sign}\begin{pmatrix} \mathbf{0} & \mathbf{A}^{1/2} \\ \mathbf{A}^{-1/2} & \mathbf{0} \end{pmatrix} \quad (119)$$

together with iterative methods for the sign function, such as the Newton–Schulz iteration, is used by default in CP2K for linear-scaling matrix inversions and (inverse) square roots of matrices.^{233,234} Several orders of iterations are available: the second-order Newton–Schulz iteration as well as third-order and fifth-order iterations based on higher-order Padé-approximants.^{224,232,235}

The sign function can also be used for the purification of the Kohn–Sham matrix \mathbf{K} into the density matrix \mathbf{P} , i.e., via the relation

$$\mathbf{P} = \frac{1}{2}(\mathbf{I} - \text{sign}(\mathbf{S}^{-1}\mathbf{K} - \mu\mathbf{I}))\mathbf{S}^{-1}. \quad (120)$$

Within CP2K, the linear-scaling calculation of the sign-function is implemented up to seventh-order based on Padé-approximants. For example, the fifth-order iteration has the form

$$\mathbf{X}_0 = \mathbf{S}^{-1}\mathbf{K} - \mu\mathbf{I}, \quad (121a)$$

$$\mathbf{X}_{k+1} = \frac{\mathbf{X}_k}{128} (35\mathbf{X}_k^8 - 180\mathbf{X}_k^6 + 378\mathbf{X}_k^4 - 420\mathbf{X}_k^2 + 315), \quad (121b)$$

$$\lim_{k \rightarrow \infty} \mathbf{X}_k = \text{sign}(\mathbf{S}^{-1}\mathbf{K} - \mu\mathbf{I}) \quad (121c)$$

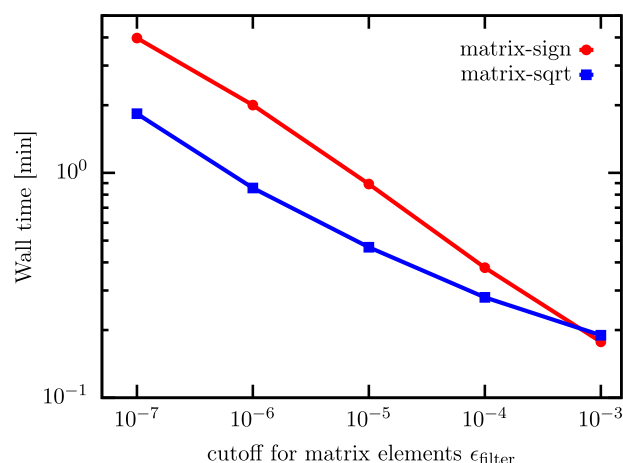


FIG. 6. Wall time for the calculation of the computationally dominating matrix-sqrt (blue, boxes) and matrix-sign (red, circles) functions as a function of matrix truncation threshold ϵ_{filter} for the STMV virus. The latter contains more than one million atoms and was simulated using the periodic implementation in CP2K of the GFN-xTB model.²³⁶ The fifth-order sign-function iteration of Eq. (121a) and the third-order sqrt-function iterations have been used. The calculations have been performed with 256 nodes (10 240 CPU-cores) of the Noctua system at the Paderborn Center for Parallel Computing (PC²).

and is implemented in CP2K with just four matrix multiplications per iteration. After each matrix multiplication, all matrix elements smaller than a threshold ϵ_{filter} are flushed to zero to retain sparsity. The scaling of the wall-clock time for the computation of the sign- and sqrt-functions to simulate the STMV virus in water solution using the Geometry, Frequency, Noncovalent, eXtended Tight-Binding (GFN-xTB) Hamiltonian is shown in Fig. 6.¹⁹⁵ The drastic speed-up of the calculation, when increasing the threshold ϵ_{filter} , immediately suggests the combination of sign-matrix iteration based linear-scaling DFT algorithms with the ideas of approximate computing (AC), as discussed in Sec. XV B.

Recently, a new iterative scheme for the inverse p-th of a matrix has been developed, which also allows us to directly evaluate the density matrix via the sign function in Eq. (118).²²⁴ An arbitrary-order implementation of this scheme is also available in CP2K.

F. Submatrix method

In addition to the sign method, the submatrix method presented in Ref. 222 has been implemented in CP2K as an alternative approach to calculate the density matrix \mathbf{P} from the KS matrix \mathbf{K} . Instead of operating on the entire matrix, calculations are performed on principal submatrices thereof. Each of these submatrices covers a set of atoms that originates from the same block of the KS matrix in the DBCSR-format, as well as those atoms in their direct neighborhood whose basis functions are overlapping. As a result, the submatrices are much smaller than the KS matrix but relatively dense. For large systems, the size of the submatrices is independent of the overall system size so that a linear-scaling method immediately results from this construction. Purification of the submatrices can be performed either using iterative schemes to compute the sign

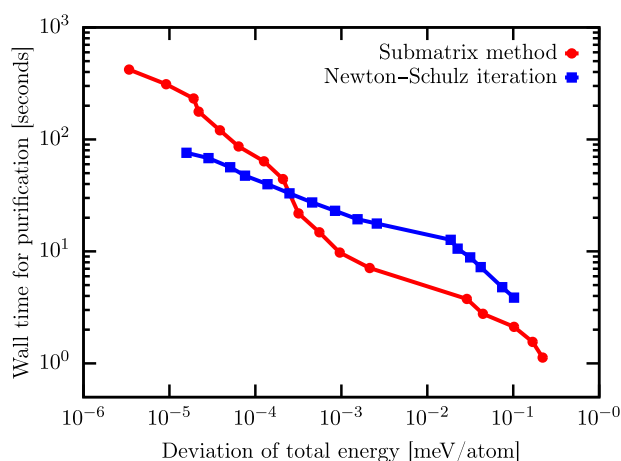


FIG. 7. Comparison of the wall time required for the purification using the submatrix method (red circles, with a direct eigendecomposition for the density matrix of the submatrices) and the second order Newton–Schulz sign iteration (blue boxes) for different relative errors in the total energy after one SCF iteration. The corresponding reference energy has been computed by setting $\epsilon_{\text{filter}} = 10^{-16}$. All calculations have been performed on a system consisting of 6192 H₂O molecules, using KS-DFT together with a SZV basis set, utilizing two compute nodes (80 CPU-cores) of the “Noctua” system at PC².

function (see Sec. VIII E) or via a direct eigendecomposition. The submatrix method provides an approximation of the density matrix \mathbf{P} , whose quality and computational cost depend on the truncation threshold ϵ_{filter} used during the SCF iterations. Figure 7 compares the accuracy provided and wall time required by the submatrix method with the accuracy and wall time of a Newton–Schulz sign iteration.

IX. LOCALIZED MOLECULAR ORBITALS

Spatially localized molecular orbitals (LMOs), also known as MLWFs in solid state physics and materials science,^{165,167} are widely used to visualize chemical bonding between atoms, help classify bonds, and thus understand electronic structure origins of observed properties of atomistic systems (see Sec. XI A). Furthermore, localized orbitals are a key ingredient in multiple local electronic structure methods that dramatically reduce the computational cost of modeling electronic properties of large atomistic systems. LMOs are also important to many other electronic structure methods that require local states such as XAS spectra modeling or dispersion-corrected XC functionals based on atomic polarizabilities.

A. Localization of orthogonal and non-orthogonal molecular orbitals

CP2K offers a variety of localization methods, in which LMOs $|j\rangle$ are constructed by finding a unitary transformation \mathbf{A} of canonical MOs $|i_0\rangle$, either occupied or virtual, i.e.,

$$|j\rangle = \sum_i |i_0\rangle A_{ij}, \quad (122)$$

which minimizes the spread of individual orbitals. CP2K uses the localization functional proposed by Resta,¹⁵⁶ which was generalized by Berghold *et al.* to periodic cells of any shape and symmetry,

$$\Omega_L(\mathbf{A}) = - \sum_K \sum_i \omega_K |z_i^K|^2, \quad (123a)$$

$$z_i^K = \sum_{mn} A_{mi} B_{mn}^K A_{ni}, \quad (123b)$$

$$B_{mn}^K = \langle m_0 | e^{i\mathbf{G}_K \cdot \hat{\mathbf{r}}} | n_0 \rangle, \quad (123c)$$

where $\hat{\mathbf{r}}$ is the position operator in three dimensions, and ω_K and \mathbf{G}_K are suitable sets of weights and reciprocal lattice vectors, respectively.¹⁶⁶ The functional in Eq. (123a) can be used for both gas-phase and periodic systems. In the former case, the functional is equivalent to the Boys–Foster localization.²³⁷ In the latter case, its applicability is limited to the electronic states described within the Γ -point approximation.

CP2K also implements the Pipek–Mezey localization functional,²³⁸ which can be written in the same form as the Berghold functional above with K referring to atoms, z_i^K measuring the contribution of orbital i to the Mulliken charge of atom K , and \mathbf{B} being defined as

$$B_{mn}^K = \frac{1}{2} \sum_{\mu \in K} \langle m_0 | (|\chi_\mu\rangle \langle \chi^\mu| + |\chi^\mu\rangle \langle \chi_\mu|) | n_0 \rangle, \quad (124)$$

where $|\chi_\mu\rangle$ and $|\chi^\mu\rangle$ are atom-centered covariant and contravariant basis set functions.¹⁶⁶ The Pipek–Mezey functional has the advantage of preserving the separation of σ and π bonds and is commonly employed for molecular systems.

In addition to the traditional localization techniques, CP2K offers localization methods that produce non-orthogonal LMOs (NLMOs).²³⁹ In these methods, matrix \mathbf{A} is not restricted to be unitary and the minimized objective function contains two terms: a localization functional Ω_L given by Eq. (123a) and a term that penalizes unphysical states with linearly dependent localized orbitals

$$\Omega(\mathbf{A}) = \Omega_L(\mathbf{A}) - c_P \log \det[\sigma_{\text{diag}}^{-1}(\mathbf{A}) \sigma(\mathbf{A})], \quad (125)$$

where $c_P > 0$ is the penalty strength and σ is the NLMO overlap matrix,

$$\sigma_{kl} = \langle k | l \rangle = \sum_{ji} A_{jk} \langle j_0 | i_0 \rangle A_{il}. \quad (126)$$

The penalty term varies from 0 for orthogonal LMOs to $+\infty$ for linearly dependent NLMOs, making the latter inaccessible in the localization procedure with finite penalty strength c_P . The inclusion of the penalty term converts the localization procedure into a straightforward unconstrained optimization problem and produces NLMOs that are noticeably more localized than their conventional orthogonal counterparts (Fig. 8).

B. Linear scaling methods based on localized one-electron orbitals

Linear-scaling, or so-called $\mathcal{O}(N)$, methods described in Sec. VIII E exploit the natural locality of the one-electron density

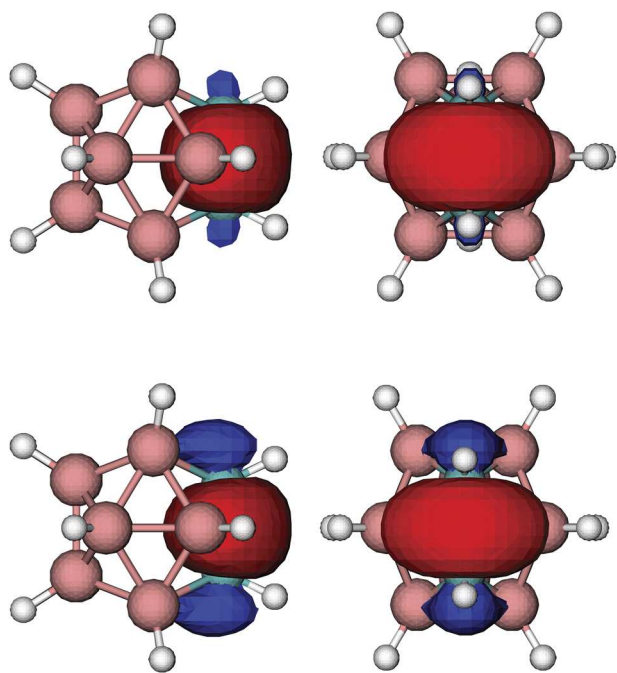


FIG. 8. Orthogonal (bottom) and non-orthogonal (top) LMOs on the covalent bond of the adjacent carbon atoms in a carborane molecule $C_2B_{10}H_{12}$ (isosurface value is 0.04 a.u.).

matrix. Unfortunately, the variational optimization of the density matrix is inefficient if calculations require many basis functions per atom. From the computational point of view, the variation of localized one-electron states is preferable to the density matrix optimization because the former requires only the occupied states, reducing the number of variational degrees of freedom significantly, especially in calculations with large basis sets. CP2K contains a variety of orbital-based $\mathcal{O}(N)$ DFT methods briefly reviewed here and in Sec. X C.

Unlike density matrices, one-electron states tend to delocalize in the process of unconstrained optimization and their locality must be explicitly enforced to achieve linear-scaling. To this end, each occupied orbital is assigned a *localization center*—an atom (or a molecule)—and a *localization radius* R_c . Then, each orbital is expanded strictly in terms of subsets of localized basis functions centered on the atoms (or molecules) lying within R_c from the orbital's center. In CP2K, contracted Gaussian functions are used as the localized basis set.

Since their introduction,^{240–242} the orbitals with this strict *a priori* localization have become known under different names including absolutely localized molecular orbitals,²⁴³ localized wavefunctions,²⁴⁴ non-orthogonal generalized Wannier functions,²⁴⁵ multi-site support functions,²⁴⁶ and non-orthogonal localized molecular orbitals.²⁴⁷ Here, they are referred to as compact localized molecular orbitals (CLMOs) to emphasize that their expansion coefficients are set to zero for all basis functions centered outside orbitals' localization subsets. Unlike previous works,^{248–250} the

ALMO acronym is avoided,²⁴³ since it commonly refers to a special case of compact orbitals with $R_c = 0$.^{180,251–256}

While the localization constraints are necessary to design orbital-based $\mathcal{O}(N)$ methods, the reduced number of electronic degrees of freedom results in the variationally optimal CLMO energy being always higher than the reference energy of fully delocalized orbitals. From the physical point of view, enforcing orbital locality prohibits the flow of electron density between distant centers and thus switches off the stabilizing donor–acceptor (i.e., covalent) component of interactions between them. It is important to note that the other interactions such as long-range electrostatics, exchange, polarization, and dispersion are retained in the CLMO approximation. Thereby, the accuracy of the CLMO-based calculations depends critically on the chosen localization radii, which should be tuned for each system to obtain the best accuracy–performance compromise. In systems with non-vanishing bandgaps, the neglected donor–acceptor interactions are typically short-ranged and CLMOs can accurately represent their electronic structure if R_c encompasses the nearest and perhaps next nearest neighbors. On the other hand, the CLMO approach is not expected to be practical for metals and semi-metals because of their intrinsically delocalized electrons.

The methods and algorithms in CP2K are designed to circumvent the known problem of slow variational optimization of CLMOs,^{244,257–261} the severity of which rendered early orbital-based $\mathcal{O}(N)$ methods impractical. Two solutions to the convergence problem are described here. The first approach is designed for systems without strong covalent interactions between localization centers such as ionic materials or ensembles of small weakly interacting molecules.^{248,249,260,262} The second approach is proposed to deal with more challenging cases of strongly bonded atoms such as covalent crystals.

The key idea of the first approach is to optimize CLMOs in a two-stage SCF procedure. In the first stage, R_c is set to zero and the CLMOs—they can be called ALMOs in this case—are optimized on their centers. In the second stage, the CLMOs are relaxed to allow delocalization onto the neighbor molecules within their localization radius R_c . To achieve a robust optimization in the problematic second stage, the delocalization component of the trial CLMOs must be kept orthogonal to the fixed ALMOs obtained in the first stage. If the delocalization is particularly weak, the CLMOs in the second stage can be obtained using the simplified Harris functional²⁶³—orbital optimization without updating the Hamiltonian matrix—or non-iterative perturbation theory. The mathematical details of the two-stage approach can be found in Ref. 249. A detailed description of the algorithms is presented in the supplementary material of Ref. 250.

The two-stage SCF approach resolves the convergence problem only if the auxiliary ALMOs resemble the final variationally optimal CLMOs and, therefore, is not practical for systems with noticeable electron delocalization—in other words, covalent bonds—between atoms. The second approach, designed for systems of covalently bonded atoms, utilizes an approximate electronic Hessian that is inexpensive to compute and yet sufficiently accurate to identify and remove the problematic optimization modes.²⁶⁴ The accuracy and practical utility of this approach rely on the fact that the removed low-curvature modes are associated with the nearly invariant occupied–occupied orbital mixing, which produce only an insignificant lowering in the total energy.

The robust variational optimization of CLMOs combined with CP2K's fast $\mathcal{O}(N)$ algorithms for the construction of the KS Hamiltonian enabled the implementation of a series of $\mathcal{O}(N)$ orbital-based DFT methods with a low computational overhead. Figure 9 shows that R_c can be tuned to achieve substantial computational savings without compromising the accuracy of the calculations. It also demonstrates that CLMO-based DFT exhibits early-offset

linear-scaling behavior even for challenging condensed-phase systems and works extremely well with large basis sets.

The current implementation of the CLMO-based methods is limited to localization centers with closed-shell electronic configurations. The nuclear gradients are available only for the methods that converge the CLMO variational optimization. This excludes CLMO methods based on the Harris functional and perturbation theory.

Section X C describes how the CLMO methods can be used in AIMD simulations by means of the second-generation Car–Parrinello MD (CPMD) method of Kühne and co-workers.^{3,265,266} Energy decomposition analysis (EDA) methods that exploit the locality of compact orbitals to understand the nature of intermolecular interactions in terms of physically meaningful components are described in Sec. XI A.

C. Polarized atomic orbitals from machine learning

The computational cost of a DFT calculation depends critically on the size and condition number of the employed basis set. Traditional basis sets are atom centered, static, and isotropic. Since molecular systems are never isotropic, it is apparent that isotropic basis sets are sub-optimal. The polarized atomic orbitals from the machine learning (PAO-ML) scheme provide small adaptive basis sets, which adjust themselves to the local chemical environment.²⁶⁷ The scheme uses polarized atomic orbitals (PAOs), which are constructed from a larger primary basis function as introduced by Berghold *et al.*²⁶⁸ A PAO basis function $\tilde{\varphi}_\mu$ can be written as a weighted sum of primary basis functions φ_ν , where μ and ν belong to the same atom,

$$\tilde{\varphi}_\mu = \sum_\nu B_{\mu\nu} \varphi_\nu. \quad (127)$$

The aim of the PAO-ML method is to predict the transformation matrix \mathbf{B} for a given chemical environment using machine learning (ML). The analytic nature of ML models allows for the calculation of exact analytic forces as they are required for AIMD simulations. In order to train such an ML model, a set of relevant atomic motifs with their corresponding optimal PAO basis are needed. This poses an intricate non-linear optimization problem as the total energy must be minimal with respect to the transformation matrix \mathbf{B} , and the electronic density, while still obeying the Pauli principle. To this end, the PAO-ML scheme introduced an improved optimization algorithm based on the Li, Nunes, and Vanderbilt formulation for the generation of training data.²⁶⁹

When constructing the ML model, great care must be taken to ensure the invariance of the predicted PAO basis sets with respect to rotations of the simulation cell to prevent artificial torque forces. The PAO-ML scheme achieves rotational invariance by employing potentials anchored on neighboring atoms. The strength of individual potential terms is predicted by the ML model. Collectively, the potential terms form an *auxiliary atomic Hamiltonian*, whose eigenvectors are then used to construct the transformation matrix \mathbf{B} .

The PAO-ML method has been demonstrated by means of AIMD simulations of liquid water. A minimal basis set yielded structural properties in fair agreement with basis set converged results. In the best case, the computational cost was reduced by a

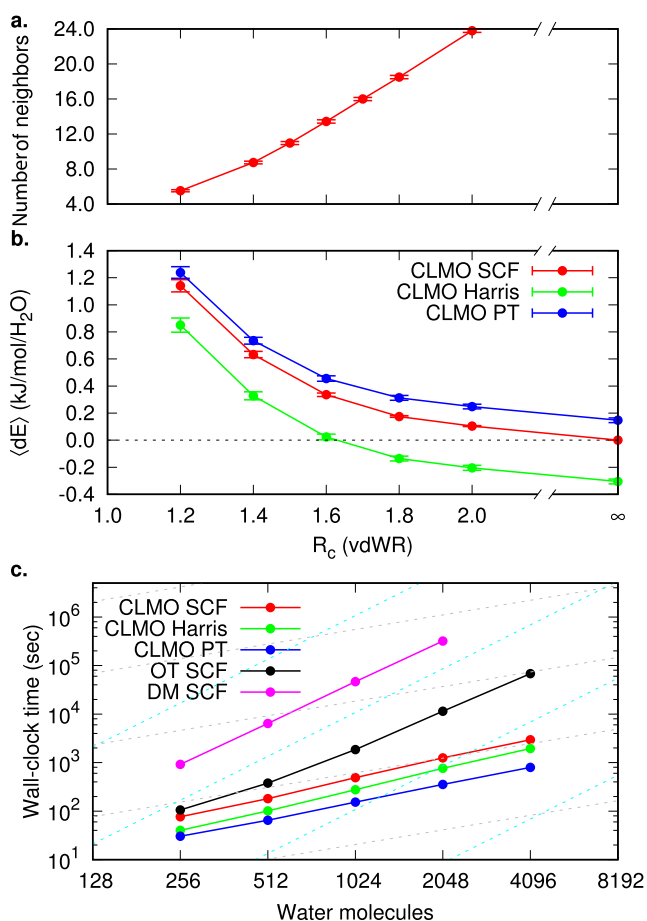


FIG. 9. Accuracy and efficiency of $\mathcal{O}(N)$ DFT for liquid water based on the two-stage CLMO optimization. Calculations were performed using the BLYP functional and a TZV2P basis set for 100 snapshots representing liquid water at constant temperature (300 K) and density (0.9966 g/cm³). (a) Dependence of the average number of neighbors on the localization radius, expressed in units of the elements' van der Waals radii (vdWR). (b) Energy error per molecule relative to the energies of fully delocalized orbitals. For simulations, in which the coordination number of molecules does not change drastically, the mean error represents a constant shift of the potential energy surface and does not affect the quality of simulations. In such cases, the standard deviation (error bars) is better suited to measure the accuracy of the CLMO methods. (c) Wall-time required for the variational minimization of the energy on 256 compute cores. The localization radius is set to 1.6 vdWR for the CLMOs methods. The CLMO methods are compared to the cubically scaling optimization of delocalized orbitals (OT SCF),^{199,212} as well as to the $\mathcal{O}(N)$ optimization of density matrix.¹¹⁵ Perfect linear- and cubic-scaling lines are shown in gray and cyan, respectively. See Ref. 249 for details.

factor of 200 and the required flops were reduced by 4 orders of magnitude. Already, a very small training set gave satisfactory results as the variational nature of the method provides robustness.

X. AB INITIO MOLECULAR DYNAMICS

The mathematical task of AIMD is to evaluate the expectation value $\langle \mathcal{O} \rangle$ of an arbitrary operator $\mathcal{O}(\mathbf{R}, \mathbf{P})$ with respect to the Boltzmann distribution

$$\langle \mathcal{O} \rangle = \frac{\int d\mathbf{R} d\mathbf{P} \mathcal{O}(\mathbf{R}, \mathbf{P}) e^{-\beta E(\mathbf{R}, \mathbf{P})}}{\int d\mathbf{R} d\mathbf{P} e^{-\beta E(\mathbf{R}, \mathbf{P})}}, \quad (128)$$

where \mathbf{R} and \mathbf{P} are the nuclear positions and momenta, while $\beta = 1/k_B T$ is the inverse temperature. The total energy function

$$E(\mathbf{R}, \mathbf{P}) = \sum_{I=1}^N \frac{\mathbf{P}_I^2}{2M_I} + E[\{\psi_i\}; \mathbf{R}], \quad (129)$$

where the first term denotes the nuclear kinetic energy, $E[\{\psi_i\}; \mathbf{R}]$ denotes the potential energy function, N denotes the number of nuclei, and M_I denotes the corresponding masses.

However, assuming the ergodicity hypothesis, the thermal average $\langle \mathcal{O} \rangle$ can not only be determined as the ensemble average but also as a temporal average,

$$\langle \mathcal{O} \rangle = \lim_{T \rightarrow \infty} \frac{1}{T} \int dt \mathcal{O}(\mathbf{R}(t), \mathbf{P}(t)), \quad (130)$$

by means of AIMD.

In the following, we will assume that the potential energy function is calculated on the fly using KS-DFT so that $E[\{\psi_i\}; \mathbf{R}] = E^{\text{KS}}[\{\psi_i[\rho(\mathbf{r})]\}; \mathbf{R}] + E_{\text{II}}(\mathbf{R})$. In CP2K, AIMD comes in two distinct flavors, which are both outlined in this section.

A. Born–Oppenheimer molecular dynamics

In Born–Oppenheimer MD (BOMD), the potential energy $E[\{\psi_i\}; \mathbf{R}]$ is minimized at every AIMD step with respect to $\{\psi_i(\mathbf{r})\}$ under the holonomic orthonormality constraint $\langle \psi_i(\mathbf{r}) | \psi_j(\mathbf{r}) \rangle = \delta_{ij}$. This leads to the following Lagrangian:

$$\begin{aligned} \mathcal{L}_{\text{BO}}(\{\psi_i\}; \mathbf{R}, \dot{\mathbf{R}}) = & \frac{1}{2} \sum_{I=1}^N M_I \dot{\mathbf{R}}_I^2 - \min_{\{\psi_i\}} E[\{\psi_i\}; \mathbf{R}] \\ & + \sum_{ij} \Lambda_{ij} (\langle \psi_i | \psi_j \rangle - \delta_{ij}), \end{aligned} \quad (131)$$

where Λ is a Hermitian Lagrangian multiplier matrix. By solving the corresponding Euler–Lagrange equations

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{\mathbf{R}}_I} = \frac{\partial \mathcal{L}}{\partial \mathbf{R}_I}, \quad (132a)$$

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \langle \dot{\psi}_i |} = \frac{\partial \mathcal{L}}{\partial \langle \psi_i |}, \quad (132b)$$

one obtains the associated equations of motion

$$\begin{aligned} M_I \ddot{\mathbf{R}}_I = & -\nabla_{\mathbf{R}_I} \left[\min_{\{\psi_i\}} E[\{\psi_i\}; \mathbf{R}] \right]_{\{\langle \psi_i | \psi_j \rangle = \delta_{ij}\}} \\ = & -\frac{\partial E}{\partial \mathbf{R}_I} + \sum_{ij} \Lambda_{ij} \frac{\partial}{\partial \mathbf{R}_I} \langle \psi_i | \psi_j \rangle \\ & - 2 \sum_i \frac{\partial \langle \psi_i |}{\partial \mathbf{R}_I} \left[\frac{\delta E}{\delta \langle \psi_i |} - \sum_j \Lambda_{ij} | \psi_j \rangle \right], \end{aligned} \quad (133a)$$

$$\begin{aligned} 0 \lesssim & -\frac{\delta E}{\delta \langle \psi_i |} + \sum_j \Lambda_{ij} | \psi_j \rangle \\ = & -\hat{H}_e \langle \psi_i | + \sum_j \Lambda_{ij} | \psi_j \rangle. \end{aligned} \quad (133b)$$

The first term on the right-hand side (RHS) of Eq. (133a) is the so-called Hellmann–Feynman force.^{270,271} The second term that is denoted as Pulay,¹⁸ or wavefunction force \mathbf{F}_{WF} , is a constraint force due to the holonomic orthonormality constraint and is nonvanishing if, and only if, the basis functions ϕ_j explicitly depend on \mathbf{R} . The final term stems from the fact that, independently of the particular basis set used, there is always an implicit dependence on the atomic positions. The factor 2 in Eq. (133a) stems from the assumption that the KS orbitals are real, an inessential simplification. Nevertheless, the whole term vanishes whenever $\psi_i(\mathbf{R})$ is an eigenfunction of the Hamiltonian within the subspace spanned by the not necessarily complete basis set.^{272,273} Note that this is a much weaker condition than the original Hellmann–Feynman theorem, of which we hence have not availed ourselves throughout the derivation, except as an eponym for the first RHS term of Eq. (133a). However, as the KS functional is nonlinear, eigenfunctions of its Hamiltonian \hat{H}_e are only obtained at exact self-consistency, which is why the last term of Eq. (133a) is also referred to as non-self-consistent force \mathbf{F}_{NSC} .²⁷⁴ Unfortunately, this cannot be assumed in any numerical calculation and results in immanent inconsistent forces as well as the inequality of Eq. (133b). Neglecting either \mathbf{F}_{WF} or \mathbf{F}_{NSC} , i.e., applying the Hellmann–Feynman theorem to a non-eigenfunction, leads merely to a perturbative estimate of the generalized forces

$$\mathbf{F} = \mathbf{F}_{\text{HF}} + \mathbf{F}_{\text{WF}} + \mathbf{F}_{\text{NSC}}, \quad (134)$$

which, contrary to the energies, depends just linearly on the error in the electronic charge density. That is why it is much more exacting to calculate accurate forces rather than total energies.

B. Second-generation Car–Parrinello molecular dynamics

Until recently, AIMD has mostly relied on two general methods: The original CPMD and the direct BOMD approach, each with its advantages and shortcomings. In BOMD, the total energy of a system, as determined by an arbitrary electronic structure method, is fully minimized in each MD time step, which renders this scheme computationally very demanding.²⁷⁵ By contrast, the original CPMD technique obviates the rather time-consuming iterative energy minimization by considering the electronic degrees of freedom as classical time-dependent fields with a fictitious mass μ that are propagated by the use of Newtonian dynamics.²⁷⁶ In order to keep the electronic and nuclear subsystems adiabatically separated, which

causes the electrons to follow the nuclei very close to their instantaneous electronic ground state, μ has to be chosen sufficiently small.²⁷⁷ However, in CPMD, the maximum permissible integration time step scales like $\sim \sqrt{\mu}$ and therefore has to be significantly smaller than that of BOMD, hence limiting the attainable simulation timescales.²⁷⁸

The so-called second-generation CPMD method combines the best of both approaches by retaining the large integration time steps of BOMD while, at the same time, preserving the efficiency of CPMD.^{3,265,266} In this Car–Parrinello-like approach to BOMD, the original fictitious Newtonian dynamics of CPMD for the electronic degrees of freedom is substituted by an improved coupled electron–ion dynamics that keeps the electrons very close to the instantaneous ground-state without the necessity of an additional fictitious mass parameter. The superior efficiency of this method originates from the fact that only one preconditioned gradient computation is necessary per AIMD step. In fact, a gain in efficiency between one and two orders of magnitude has been observed for a large variety of different systems ranging from molecules and solids,^{279–287} including phase-change materials,^{288–294} over aqueous systems,^{295–304} to complex interfaces.^{305–311}

Within mean-field electronic structure methods, such as Hartree–Fock and KS-DFT, $E[\{\psi_i\}; \mathbf{R}]$ is either a functional of the electronic wavefunction that is described by a set of occupied MOs $|\psi_i\rangle$ or, equivalently, of the corresponding one-particle density operator $\rho = \sum_i |\psi_i\rangle\langle\psi_i|$. The improved coupled electron–ion dynamics of second-generation CPMD obeys the following equations of motion for the nuclear and electronic degrees of freedom:

$$\begin{aligned} M_I \ddot{\mathbf{R}}_I &= -\nabla_{\mathbf{R}_I} \left[\min_{\{\psi_i\}} E[\{\psi_i\}; \mathbf{R}_I] \right]_{\{\langle\psi_i|\psi_j\rangle=\delta_{ij}\}} \\ &= -\frac{\partial E}{\partial \mathbf{R}_I} + \sum_{ij} \Lambda_{ij} \frac{\partial}{\partial \mathbf{R}_I} \langle\psi_i|\psi_j\rangle \\ &\quad - 2 \sum_i \frac{\partial \langle\psi_i|}{\partial \mathbf{R}_I} \left[\frac{\partial E[\{\psi_i\}; \mathbf{R}_I]}{\partial \langle\psi_i|} - \sum_j \Lambda_{ij} |\psi_j\rangle \right], \quad (135a) \end{aligned}$$

$$\begin{aligned} \frac{d^2}{d\tau^2} |\psi_i(\mathbf{r}, \tau)\rangle &= -\frac{\partial E[\{\psi_i\}; \mathbf{R}_I]}{\partial \langle\psi_i(\mathbf{r}, \tau)|} - \gamma \omega \frac{d}{d\tau} |\psi_i(\mathbf{r}, \tau)\rangle \\ &\quad + \sum_j \Lambda_{ij} |\psi_j(\mathbf{r}, \tau)\rangle. \quad (135b) \end{aligned}$$

The former is the conventional nuclear equation of motion of BOMD consisting of Hellmann–Feynman, Pulay, and non-self-consistent force contributions,^{18,270,271,274} whereas the latter constitutes a universal oscillator equation as obtained by nondimensionalization. The first term on the RHS of Eq. (135b) can be sought of as an “electronic force” to propagate $|\psi_i\rangle$ in dimensionless time τ . The second term is an additional damping term to relax more quickly to the instantaneous electronic ground-state, where γ is an appropriate damping coefficient and ω is the undamped angular frequency of the universal oscillator. The final term derives from the constraint to fulfill the holonomic orthonormality condition $\langle\psi_i|\psi_j\rangle = \delta_{ij}$, by using the Hermitian Lagrangian multiplier matrix Λ . As shown in Eq. (135b), not even a single diagonalization step but just one “electronic force” calculation is required. In other words, the second-generation CPMD method not only entirely bypasses the

necessity of a SCF cycle but also the alternative iterative wavefunction optimization.

However, contrary to the evolution of the nuclei, for the short-term integration of the electronic degrees of freedom, accuracy is crucial, which is why a highly accurate yet efficient propagation scheme is essential. As a consequence, the evolution of the MOs is conducted by extending the always-stable predictor–corrector integrator of Kolafa to the electronic structure problem.³¹² However, since this scheme was originally devised to deal with classical polarization, special attention must be paid to the fact that the holonomic orthonormality constraint, which is due to the fermionic nature of electrons that forces the wavefunction to be antisymmetric in order to comply with the Pauli exclusion principle, is always satisfied during the dynamics. For that purpose, first the predicted MOs at time t_n are computed based on the electronic degrees of freedom from the K previous AIMD steps,

$$|\psi_i^p(t_n)\rangle = \sum_m^K (-1)^{m+1} m \frac{\binom{2K}{K-m}}{\binom{2K-2}{K-1}} \rho(t_{n-m}) |\psi_i(t_{n-1})\rangle. \quad (136)$$

This is to say that the predicted one-electron density operator $\rho^p(t_n)$ is used as a projector onto the occupied subspace $|\psi_i(t_{n-1})\rangle$ of the previous AIMD step. In this way, we take advantage of the fact that $\rho^p(t_n)$ evolves much more smoothly than $|\psi_i^p(t_n)\rangle$ and is therefore easier to predict. This is particularly true for metallic systems, where many states crowd the Fermi level. Yet, to minimize the error and to further reduce the deviation from the instantaneous electronic ground-state, $|\psi_i^p(t_n)\rangle$ is corrected by performing a single minimization step $|\delta\psi_i^p(t_n)\rangle$ along the preconditioned electronic gradient direction, as computed by the orthonormality conserving orbital OT method described in Sec. VIII C.¹⁹⁹ Therefore, the modified corrector can be written as

$$|\psi_i(t_n)\rangle = |\psi_i^p(t_n)\rangle + \omega (|\delta\psi_i^p(t_n)\rangle - |\psi_i^p(t_n)\rangle)$$

with

$$\omega = \frac{K}{2K-1} \text{ for } K \geq 2. \quad (137)$$

The eventual predictor–corrector scheme leads to an electron dynamics that is rather accurate and time-reversible up to $\mathcal{O}(\Delta t^{2K-2})$, where Δt is the discretized integration time step, while ω is chosen so as to guarantee a stable relaxation toward the minimum. In other words, the efficiency of the present second-generation CPMD method stems from the fact that the instantaneous electronic ground state is very closely approached within just one electronic gradient calculation. We thus totally avoid the SCF cycle and any expensive diagonalization steps while remaining very close to the BO surface and, at the same time, Δt can be chosen to be as large as in BOMD.

However, despite the close proximity of the propagated MOs to the instantaneous electronic ground state, the nuclear dynamics is slightly dissipative, most likely because the employed predictor–corrector scheme is not symplectic. The validity of this assumption has been independently verified by various numerical studies of others.^{2,313–315} Nevertheless, presuming that the energy is exponentially decaying, which had been shown to be an excellent

assumption,^{3,265,266} it is possible to rigorously correct for the dissipation by modeling the nuclear forces of second-generation CPMD $\mathbf{F}_I^{\text{CP}} = -\nabla_{\mathbf{R}_I} E[\{\psi_i\}; \mathbf{R}_I]$ by

$$\mathbf{F}_I^{\text{CP}} = \mathbf{F}_I^{\text{BO}} - \gamma_D M_I \dot{\mathbf{R}}_I, \quad (138)$$

where \mathbf{F}_I^{BO} are the exact but inherently unknown BO forces and γ_D is an intrinsic yet to be determined friction coefficient to mimic the dissipation. The presence of damping immediately suggests a canonical sampling of the Boltzmann distribution by means of the following modified Langevin-type equation:

$$M_I \ddot{\mathbf{R}}_I = \underbrace{\mathbf{F}_I^{\text{BO}} - \gamma_D M_I \dot{\mathbf{R}}_I}_{\mathbf{F}_I^{\text{CP}}} + \boldsymbol{\Xi}_I^{\text{D}} \quad (139a)$$

$$= \mathbf{F}_I^{\text{CP}} + \boldsymbol{\Xi}_I^{\text{D}}, \quad (139b)$$

where $\boldsymbol{\Xi}_I^{\text{D}}$ is an additive white noise term, which must obey the fluctuation–dissipation theorem $\langle \boldsymbol{\Xi}_I^{\text{D}}(0) \boldsymbol{\Xi}_I^{\text{D}}(t) \rangle = 2\gamma_D M_I k_B T \delta(t)$ in order to guarantee an accurate sampling of the Boltzmann distribution.

This is to say that if one knew the unknown value of γ_D , it would nevertheless be possible to ensure an exact canonical sampling of the Boltzmann distribution despite the dissipation. Fortunately, the inherent value of γ_D does not need to be known *a priori* but can be bootstrapped so as to generate the correct average temperature,^{3,265,266} as measured by the equipartition theorem

$$\left\langle \frac{1}{2} M_I \dot{\mathbf{R}}_I^2 \right\rangle = \frac{3}{2} k_B T. \quad (140)$$

More precisely, in order to determine the hitherto unknown value of γ_D , we perform a preliminary simulation in which we vary γ_D on the fly using a Berendsen-like algorithm until Eq. (140) is eventually satisfied.²⁹⁵ Alternatively, γ_D can also be continuously adjusted automatically using the adaptive Langevin technique of Leimkuhler and co-workers.^{316–318} In this method, the friction coefficient γ_D of Eq. (139) is reinterpreted as a dynamical variable, defined by a negative feedback loop control law as in the Nosé–Hoover scheme.^{319,320} The corresponding dynamical equation for γ_D reads as

$$\dot{\gamma}_D = (2K - n k_B T) / Q, \quad (141)$$

where K is the kinetic energy, n is the number of degrees of freedom, and $Q = k_B T \tau_{\text{NH}}^2$ is the Nose–Hoover fictitious mass with time constant τ_{NH} .

C. Low-cost linear-scaling *ab initio* molecular dynamics based on compact localized molecular orbitals

The computational complexity of CLMO DFT described in Sec. IX B grows linearly with the number of molecules, while its overhead cost remains very low because of the small number of electronic descriptors and efficient optimization algorithms (see Fig. 9). These features make CLMO DFT a promising method for accurate AIMD simulations of large molecular systems.

The difficulty in adopting CLMO DFT for dynamical simulations arises from the nonvariational character of compact orbitals. While CLMOs can be reliably optimized using the two-stage SCF procedure described in Sec. IX B, the occupied subspace defined in the first stage must remain fixed during the second stage to

achieve convergence. In addition, electron delocalization effects can suddenly become inactive in the course of a dynamical simulation when a neighboring molecule crosses the localization threshold R_c . Furthermore, the variational optimization of orbitals is in practice inherently not perfect and terminated once the norm of the electronic gradient drops below a small but nevertheless finite convergence threshold ϵ_{SCF} . These errors accumulate in long AIMD trajectories, leading to non-physical sampling and/or numerical problems.

Traditional strategies to cope with these problems are computationally expensive and include increasing R_c , decreasing ϵ_{SCF} , and computing the nonvariational contribution to the forces via a coupled-perturbed procedure.^{321,322} CP2K implements another approach that uses the CLMO state obtained after the two-stage CLMO SCF procedure to compute only the inexpensive Hellmann–Feynman and Pulay components and neglects the computationally intense nonvariational component of the nuclear forces. To compensate for the missing component, CP2K uses a modified Langevin equation of motion that is fine-tuned to retain stable dynamics even with imperfect forces. This approach is known as the second-generation CPMD methodology of Kühne and workers,^{3,265,266} which is discussed in detail in Sec. X B. Its combination with CLMO DFT is described in Ref. 250.

An application of CLMO AIMD to liquid water demonstrates that the compensating terms in the modified Langevin equation can be tuned to maintain a stable dynamics and reproduce accurately multiple structural and dynamical properties of water with tight orbital localization ($R_c = 1.6$ vdWR) and ϵ_{SCF} as high as 10^{-2} a.u.²⁵⁰ The corresponding results are shown in Fig. 10. The low computational overhead of CLMO AIMD, afforded by these settings, makes it ideal for routine medium-scale AIMD simulations, while its linear-scaling complexity allows us to extend first-principle studies of molecular systems to previously inaccessible length scales.

It is important to note that AIMD in CP2K cannot be combined with the CLMO methods based on the Harris functional and perturbative theory (see Sec. IX B). A generalization of the CLMO AIMD methodology to systems of strongly interacting atoms (e.g., covalent crystals) is underway.

D. Multiple-time-step integrator

The AIMD-based MTS integrator presented here is based on the reversible reference system propagator algorithm (r-RESPA), which was developed for classical MD simulations.³²³ Using a carefully constructed integration scheme, the time evolution is reversible, and the MD simulation remains accurate and energy conserving. In AIMD-MTS, the difference in computational cost between a hybrid and a local XC functional is exploited, by performing a hybrid calculation only after several conventional DFT integration time steps. r-RESPA is derived from the Liouville operator representation of Hamilton mechanics,

$$iL = \sum_{j=1}^f \left[\frac{\partial H}{\partial p_j} \frac{\partial}{\partial x_j} + \frac{\partial H}{\partial x_j} \frac{\partial}{\partial p_j} \right], \quad (142)$$

where L is the Liouville operator for the system containing f degrees of freedom. The corresponding positions and momenta are denoted

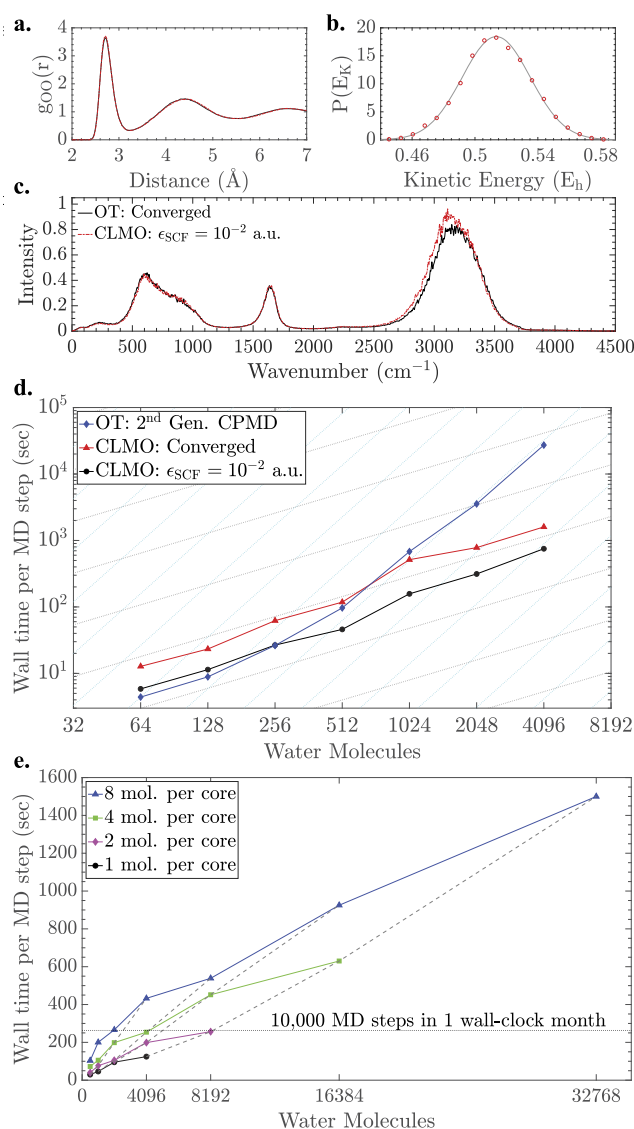


FIG. 10. Accuracy and efficiency of $\mathcal{O}(N)$ AIMD based on the two-stage CLMO SCF for liquid water. All simulations were performed using the PBE XC functional and a TZV2P basis set at constant temperature (300 K) and density (0.9966 g/cm^3). For the CLMO methods, $R_c = 1.6 \text{ vdWR}$. (a) Radial distribution function, (b) kinetic energy distribution (the gray curve shows the theoretical Maxwell-Boltzmann distribution), and (c) IR spectrum were calculated using the fully converged OT method for delocalized orbitals (black line) and CLMO AIMD with $\epsilon_{SCF} = 10^{-2}$ a.u. (d) Timing benchmarks for liquid water on 256 compute cores. Cyan lines represent perfect cubic-scaling, whereas gray lines represent perfect linear-scaling. (e) Weak scalability benchmarks are performed with $\epsilon_{SCF} = 10^{-2}$ a.u. Gray dashed lines connect systems simulated on the same number of cores to confirm $\mathcal{O}(N)$ behavior. See Ref. 250 for details.

as x_j and p_j , respectively. This operator is then used to create the classical propagator $U(t)$ for the system, which reads as

$$U(t) = e^{iLt}. \quad (143)$$

Decomposing the Liouville operator into two parts,

$$iL = iL_1 + iL_2, \quad (144)$$

and applying a second-order Trotter-decomposition to the corresponding propagator yields

$$e^{i(L_1+L_2)\Delta t} = \left[e^{i(L_1+L_2)\Delta t/n} \right]^n = \left[e^{iL_1(\delta t/2)} e^{iL_2\delta t} e^{iL_1(\delta t/2)} \right]^n + \mathcal{O}(\delta t^3), \quad (145)$$

with $\delta t = \Delta t/n$. For this propagator, several integrator schemes can be derived.³²⁴ The extension for AIMD-MTS is obtained from a decomposition of the force in the Liouville operator into two or more separate forces, i.e.,

$$iL = \sum_{j=1}^f \left[\dot{x}_j \frac{\partial}{\partial x_j} + F_j^1 \frac{\partial}{\partial p_j} + F_j^2 \frac{\partial}{\partial p_j} \right]. \quad (146)$$

For that specific case, the propagator can be written as

$$e^{iL\Delta t} = e^{(\Delta t/2)F^2 \frac{\partial}{\partial p}} \left[e^{(\delta t/2)F^1 \frac{\partial}{\partial p}} e^{\delta t \dot{x}_j \frac{\partial}{\partial x_j}} e^{(\delta t/2)F^1 \frac{\partial}{\partial p}} \right]^n e^{(\Delta t/2)F^2 \frac{\partial}{\partial p}}. \quad (147)$$

This allows us to treat F^1 and F^2 with different time steps, while the whole propagator still remains time-reversible. The procedure for F^1 and F^2 will be referred to as the inner and the outer loop, respectively. In the AIMD-MTS approach, the forces are split in the following way:

$$F^1 = F^{\text{approx}}, \quad (148a)$$

$$F^2 = F^{\text{accur}} - F^{\text{approx}}, \quad (148b)$$

where F^{accur} are the forces computed by the more accurate method, e.g., hybrid DFT, whereas F^{approx} are forces as obtained from a less accurate method, e.g., by GGA XC functionals. Obviously, the corresponding Liouville operator equals the purely accurate one. The advantage of this splitting is that the magnitude of F^2 is usually much smaller than that of F^1 . To appreciate that, it has to be considered how closely geometries and frequencies obtained by hybrid DFT normally match the ones obtained by a local XC functional, in particular, for stiff degrees of freedom. The difference of the corresponding Hessians is therefore small and low-frequent. However, high-frequency parts are not removed analytically; thus, the theoretical upper limit for the time step of the outer loop remains half the period of the fastest vibration.³²⁵ The gain originates from an increased accuracy and stability for larger time steps in the outer loop integration. Even using an outer loop time step close to the theoretical limit, a stable and accurate AIMD is obtained. Additionally, there is no shift to higher frequencies as the (outer loop) time step is increased, contrary to the single time step case.

XI. ENERGY DECOMPOSITION AND SPECTROSCOPIC ANALYSIS METHODS

Within CP2K, the nature of intermolecular bonding and vibrational spectra can be rationalized by EDA, normal mode analysis (NMA), and mode selective vibrational analysis (MSVA) methods, respectively.

A. Energy decomposition analysis based on compact localized molecular orbitals

Intermolecular bonding is a result of the interplay of electrostatic interactions between permanent charges and multipole moments on molecules, polarization effects, Pauli repulsion, donor-acceptor orbital interactions (also known as covalent, charge-transfer, or delocalization interactions), and weak dispersive forces. The goal of EDA is to measure the contribution of these components within the total binding energy, thus gaining deeper insight into physical origins of intermolecular bonds.

To that extent, CP2K contains an EDA method based on ALMOs—molecular orbitals localized entirely on single molecules or ions within a larger system.^{248,252} The ALMO EDA separates the total interaction energy (ΔE_{TOT}) into frozen density (FRZ), polarization (POL), and charge-transfer (CT) terms, i.e.,

$$\Delta E_{\text{TOT}} = \Delta E_{\text{FRZ}} + \Delta E_{\text{POL}} + \Delta E_{\text{CT}}, \quad (149)$$

which is conceptually similar to the Kitaura–Morokuma EDA³²⁶—one of the first EDA methods. The frozen interaction term is defined as the energy required to bring isolated molecules into the system without any relaxation of their MOs, apart from modifications associated with satisfying the Pauli exclusion principle,

$$\Delta E_{\text{FRZ}} \equiv E(\mathbf{R}_{\text{FRZ}}) - \sum_x E(\mathbf{R}_x), \quad (150)$$

where $E(\mathbf{R}_x)$ is the energy of isolated molecule x and \mathbf{R}_x is the corresponding density matrix, whereas \mathbf{R}_{FRZ} is the density matrix of the system constructed from the unrelaxed MOs of the isolated molecules. The ALMO EDA is also closely related to the block-localized wavefunction EDA³²⁷ because both approaches use the same variational definition of the polarization term as the energy lowering due to the relaxation of each molecule's ALMOs in the field of all other molecules in the system,

$$\Delta E_{\text{POL}} \equiv E(\mathbf{R}_{\text{ALMO}}) - E(\mathbf{R}_{\text{FRZ}}). \quad (151)$$

The strict locality of ALMOs is utilized to ensure that the relaxation is constrained to include only intramolecular variations. This approach, whose mathematical and algorithmic details have been described by many authors,^{243,248,251,328,329} gives an upper limit to the true polarization energy.³³⁰ The remaining portion of the total interaction energy, the CT term, is calculated as the difference in the energy of the relaxed ALMO state and the state of fully delocalized optimized orbitals (\mathbf{R}_{SCF}),

$$\Delta E_{\text{CT}} \equiv E(\mathbf{R}_{\text{SCF}}) - E(\mathbf{R}_{\text{ALMO}}). \quad (152)$$

A distinctive feature of the ALMO EDA is that the charge-transfer contribution can be separated into contributions associated with forward and back-donation for each pair of molecules, as well as a many-body higher-order (induction) contribution (HO), which is very small for typical intermolecular interactions. Both the amount of the electron density transferred between a pair of molecules and the corresponding energy lowering can be computed,

$$\Delta Q_{\text{CT}} = \sum_{x,y>x} \{ \Delta Q_{x \rightarrow y} + \Delta Q_{y \rightarrow x} \} + \Delta Q_{\text{HO}}, \quad (153a)$$

$$\Delta E_{\text{CT}} = \sum_{x,y>x} \{ \Delta E_{x \rightarrow y} + \Delta E_{y \rightarrow x} \} + \Delta E_{\text{HO}}. \quad (153b)$$

The ALMO EDA implementation in CP2K is currently restricted to closed-shell fragments. The efficient $\mathcal{O}(N)$ optimization of ALMOs serves as its underlying computational engine.²⁴⁹ The ALMO EDA in CP2K can be applied to both gas-phase and condensed matter systems. It has been recently extended to fractionally occupied ALMOs,³³¹ thus enabling the investigation of interactions between metal surfaces and molecular adsorbates. Another unique feature of the implementation in CP2K is the ability to control the spatial range of charge-transfer between molecules using the cutoff radius R_c of CLMOs (see Sec. IX B). Additionally, the ALMO EDA in combination with CP2K's efficient AIMD engine allows us to switch off the CT term in AIMD simulations, thus measuring the CT contribution to dynamical properties of molecular systems.³³²

The ALMO EDA has been applied to study intermolecular interactions in a variety of gas- and condensed-phase molecular systems.^{331,333} The CP2K implementation of ALMO EDA has been particularly advantageous to understand the nature of hydrogen bonding in bulk liquid water,^{180,248,254,297,300,301,303,332} ice,³³⁴ and water phases confined to low dimensions.³³⁵ Multiple studies have pointed out a deep connection between the donor-acceptor interactions and features of the x-ray absorption,^{254,336} infrared (IR)^{297,298,300,303,332,337,338} (Fig. 11), and nuclear magnetic resonance

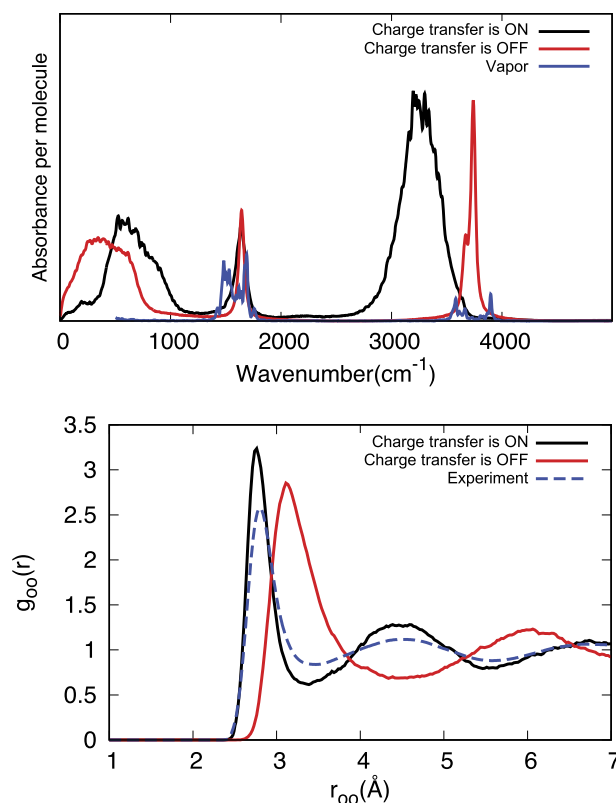


FIG. 11. Comparison of the calculated radial distribution functions and IR spectra computed for liquid water based on AIMD simulations at the BLYP + D3/TZV2P level of theory with and without the CT terms. See Ref. 332 for details.

spectra of liquid water.¹⁸⁰ Extension of ALMO EDA to AIMD simulations has shown that the insignificant covalent component of HB determines many unique properties of liquid water including its structure (Fig. 11), dielectric constant, hydrogen bond lifetime, rate of self-diffusion, and viscosity.^{250,332}

B. Normal mode analysis of infrared spectroscopy

IR vibrational spectra can be obtained with CP2K by carrying out a NMA, within the Born–Oppenheimer approximation. The expansion of the potential energy in a Taylor series around the equilibrium geometry gives

$$E_{\text{pot}}(\{\mathbf{R}_A\}) = E_{\text{pot}}^{(0)} + \sum_A \left(\frac{\partial E_{\text{pot}}}{\partial \mathbf{R}_A} \right) \delta \mathbf{R}_A + \frac{1}{2} \sum_{AB} \left(\frac{\partial^2 E_{\text{pot}}}{\partial \mathbf{R}_A \partial \mathbf{R}_B} \right) \delta \mathbf{R}_A \delta \mathbf{R}_B + \dots \quad (154)$$

At the equilibrium geometry, the first derivative is zero by definition; hence, in the harmonic approximation, only the second derivative matrix (Hessian) has to be calculated. In our implementation, the Hessian is obtained with the three-point central difference approximation, which for a system of M particles corresponds to $6M$ force evaluations. The Hessian matrix is diagonalized to determine the $3M$ \mathcal{Q} vectors representing a set of decoupled coordinates, which are the normal modes. In mass weighted coordinates, the angular frequencies of the corresponding vibrations are then obtained by the eigenvalues of the second derivative matrix.

C. Mode selective vibrational analysis

For applications in which only few vibrations are of interest, the MSVA presents a possibility to reduce the computational cost significantly.^{339,340} Instead of computing the full second derivative matrix, only a subspace is evaluated iteratively using the Davidson algorithm. As in NMA, the starting point of this method is the eigenvalue equation

$$\mathbf{H}^{(m)} \mathcal{Q}_k = \lambda_k \mathcal{Q}_k, \quad (155)$$

where $\mathbf{H}^{(m)}$ is the mass weighted Hessian in Cartesian coordinates. Instead of numerically computing the Hessian elementwise, the action of the Hessian on a predefined or arbitrary collective displacement vector

$$\mathbf{d} = \sum_i d_i \mathbf{e}_i^m \quad (156)$$

is chosen, where $\mathbf{e}_i^{(m)}$ are the $3M$ nuclear basis vectors. The action of the Hessian on \mathbf{d} is given in the first approximation as

$$\sigma_k = (\mathbf{H}^{(m)} \mathbf{d})_k = \sum_l \frac{\partial^2 E_{\text{pot}}}{\partial \mathbf{R}_l^{(m)} \partial \mathbf{R}_k^{(m)}} d_l = \frac{\partial^2 E_{\text{pot}}}{\partial \mathbf{R}_k^{(m)} \partial \mathbf{d}}. \quad (157)$$

The first derivatives with respect to the nuclear positions are simply the force which can be computed analytically. The derivative of the force along components i with respect to d can then be evaluated as a numerical derivative using the three-point central difference approximation to yield the vector σ . The subspace

approximation to the Hessian at the i th iteration of the procedure is a $i \times i$ matrix,

$$\mathbf{H}_{\text{approx}}^{(m),i} = \mathbf{D}^T \mathbf{H}^{(m)} \mathbf{D} = \mathbf{D}^T \Sigma, \quad (158)$$

where \mathbf{B} and Σ are the collection of the \mathbf{d} and Σ vectors up to the actual iteration step. Solving the eigenvalue problem for the small Davidson matrix

$$\mathbf{H}_{\text{approx}}^{(m),i} \mathbf{u}^i = \tilde{\lambda}^i \mathbf{u}^i, \quad (159)$$

we obtain approximate eigenvalues $\tilde{\lambda}_k^i$. From the resulting eigenvectors, the residuum vector

$$\mathbf{r}_m^i = \sum_{k=1}^i u_{mk}^i [\sigma^k - \tilde{\lambda}_m^i \mathbf{d}^k], \quad (160)$$

where the sum is over all the basis vectors \mathbf{d}^k , the number of which increases at each new iteration i . The residuum vector is used as a displacement vector for the next iteration. The approximation to the exact eigenvector m at the i th iteration is

$$\mathcal{Q}_m \approx \sum_{k=1}^i u_{mk}^i \mathbf{d}^k. \quad (161)$$

This method avoids the evaluation of the complete Hessian and therefore requires fewer force evaluations. Yet, in the limit of $3M$ iterations, the exact Hessian is obtained and thus the exact frequencies and normal modes in the limit of the numerical difference approximation.

As this is an iterative procedure (Davidson subspace algorithm), the initial guess is important for convergence. Moreover, there is no guarantee that the created subspace will contain the desired modes in the case of a bad initial guess. In CP2K, the choice of the target mode can be a single mode, a range of frequencies, or modes localized on preselected atoms. If little is known about the modes of interest (e.g., a frequency range and contributing atoms), an initial guess can be built by a random displacement of atoms. In case, where a single mode is tracked, one can use normal modes obtained from lower quality methods. The residuum vector is calculated with respect to the mode with the eigenvalue closest to the input frequency. The method will only converge to the mode of interest if the initial guess is suitable. With the implemented algorithm, always the mode closest to the input frequency is improved. Using an arbitrary initial guess, the mode of interest might not be present in the subspace at the beginning. It is important to note that the Davidson algorithm might converge before the desired mode becomes part of the subspace. Therefore, there is no warranty that the algorithm would always converge to the desired mode. By giving a range of frequencies as initial guess, it might happen that either none or more than one mode is already present in the spectrum. In the first case, the mode closest to the desired range will be tracked. In the second case, always the least converged mode will be improved. If the mode is selected by a list of contributing atoms, at each step, the approximations to the eigenvectors of the full Hessian are calculated, and the vector with the largest contributions of the selected atoms is tracked for the next iteration step.

The MSVA scheme can be efficiently parallelized by either distributing the force calculations or using the block Davidson algorithm. In the latter approach, the parallel environment is split into

n sub-environments, each consisting of m processors performing the force calculations. The initial vectors are constructed for all n environments such that n orthonormal vectors are generated. After each iteration step, the new \mathbf{d} and σ vectors are combined into a single \mathbf{D} and Σ matrix. These matrices are then used to construct the approximate Hessian, and from this, the n modes closest to the selected vectors are again distributed.

The IR intensities are defined as the derivatives of the dipole vector with respect to the normal mode. Therefore, it is necessary to activate the computation of the dipoles together with the NMA. By applying the mode selective approach, large condensed matter systems can be addressed. An illustrative example is the study by Schiffmann *et al.*,³⁴¹ where the interaction of the N3, N719, and N712 dyes with anatase(101) has been modeled. The vibrational spectra for all low-energy conformations have been computed and used to assist the assignment of the experimental IR spectra, revealing a protonation dependent binding mode and the role of self-assembly in reaching high coverage.

XII. EMBEDDING METHODS

CP2K aims to provide a wide range of potential energy methods, ranging from empirical approaches such as classical force-fields over DFT-based techniques to quantum chemical methods. In addition, multiple descriptions can be arbitrarily combined at the input level so that many combinations of methods are directly available. Examples are schemes that combine two or more potential energy surfaces via

$$E[\mathbf{R}] = E^{\text{MM}}[\mathbf{R}_{I+II}] - E^{\text{MM}}[\mathbf{R}_I] + E^{\text{QM}}[\mathbf{R}_I], \quad (162)$$

linear combinations of potentials as necessary in alchemical free-energy calculations,

$$E_\lambda[\mathbf{R}] = \lambda E_I[\mathbf{R}] + (1 - \lambda) E_{II}[\mathbf{R}], \quad (163)$$

or propagation of the lowest potential energy surfaces,

$$E[\mathbf{R}] = \min[E_I[\mathbf{R}], E_{II}[\mathbf{R}], \dots]. \quad (164)$$

However, beside such rather simplistic techniques,³⁴² more sophisticated embedding methods described in the following are also available within CP2K.

A. QM/MM methods

The QM/molecular-mechanics (MM) multi-grid implementation in CP2K is based on the use of an additive mixed quantum-classical mechanics (QM/MM) scheme.^{343–345} The total energy of the molecular system can be partitioned into three disjointed terms,

$$E_{\text{TOT}}(\mathbf{R}_\alpha, \mathbf{R}_a) = E^{\text{QM}}(\mathbf{R}_\alpha) + E^{\text{MM}}(\mathbf{R}_a) + E^{\text{QM/MM}}(\mathbf{R}_\alpha, \mathbf{R}_a). \quad (165)$$

These energy terms parametrically depend on the coordinates of the nuclei in the quantum region (\mathbf{R}_α) and on classical atoms (\mathbf{R}_a). Hence, E^{QM} is the pure quantum energy, computed using the QUICKSTEP code,⁵ whereas E^{MM} is the classical energy, described through the use of the internal classical molecular-mechanics (MM) driver called FIST. The latter allows the use of the most common

force-fields employed in MM simulations.^{346,347} The interaction energy term $E^{\text{QM/MM}}$ contains all non-bonded contributions between the QM and the MM subsystems, and in a DFT framework, we express it as

$$E^{\text{QM/MM}}(\mathbf{R}_\alpha, \mathbf{R}_a) = \sum_{a \in \text{MM}} q_a \int \rho(\mathbf{r}, \mathbf{R}_\alpha) v_a(|\mathbf{r} - \mathbf{R}_a|) d\mathbf{r} + \sum_{a \in \text{MM}, \alpha \in \text{QM}} v_{\text{NB}}(\mathbf{R}_\alpha, \mathbf{R}_a), \quad (166)$$

where \mathbf{R}_a is the position of the MM atom a with charge q_a , $\rho(\mathbf{r}, \mathbf{R}_\alpha)$ is the total (electronic plus nuclear) charge density of the quantum system, and $v_{\text{NB}}(\mathbf{R}_\alpha, \mathbf{R}_a)$ is the non-bonded interaction between classical atom a and quantum atom α . The electrostatic potential (ESP) of the MM atoms $v_a(|\mathbf{r} - \mathbf{R}_a|)$ is described using for each atomic charge a Gaussian charge distribution,

$$\rho(|\mathbf{r} - \mathbf{R}_a|) = \left(\frac{1}{\sqrt{\pi} r_{c,a}} \right)^3 \exp(-(|\mathbf{r} - \mathbf{R}_a|/r_{c,a})^2), \quad (167)$$

with width $r_{c,a}$, eventually resulting in

$$v_a(|\mathbf{r} - \mathbf{R}_a|) = \frac{\text{erf}(|\mathbf{r} - \mathbf{R}_a|/r_{c,a})}{|\mathbf{r} - \mathbf{R}_a|}. \quad (168)$$

This renormalized potential has the desired property of tending to $1/r$ at large distances and going smoothly to a constant for small r (see Ref. 348 for renormalization details). Due to the Coulomb long-range behavior, the computational cost of the integral in Eq. (166) can be very large. In CP2K, we designed a decomposition of the electrostatic potential in terms of Gaussian functions with different cutoffs combined with a real-space multi-grid framework to accelerate the calculation of the electrostatic interaction. We named this method Gaussian Expansion of the Electrostatic Potential or GEEP.³⁴⁸ The advantage of this methodology is that grids of different spacing can be used to represent the different contributions of $v_a(\mathbf{r}, \mathbf{R}_a)$, instead of using only the same grid employed for the mapping of the electronic wavefunction. In fact, by writing a function as a sum of terms with compact support and with different cutoffs, the mapping of the function can be efficiently achieved using different grid levels, in principle as many levels as contributing terms, each optimal to describe the corresponding term.

1. QM/MM for isolated systems

For isolated systems, each MM atom is represented as a continuous Gaussian charge distribution and each GEEP term is mapped on one of the available grid levels, chosen to be the first grid whose cutoff is equal to or bigger than the cutoff of that particular GEEP contribution. However, all atoms contribute to the coarsest grid level through the long-range $R_{\text{low}}(|\mathbf{r} - \mathbf{R}_a|)$ part,³⁴⁸ which is the smooth GEEP component as defined in Eq. (169). The result of this collocation procedure is a multi-grid representation of the QM/MM electrostatic potential $V_i^{\text{QM/MM}}(\mathbf{r}, \mathbf{R}_a)$, where i labels the grid level, represented by a sum of single atomic contributions $V_i^{\text{QM/MM}}(\mathbf{r}, \mathbf{R}_a) = \sum_{a \in \text{MM}} v_a^i(\mathbf{r}, \mathbf{R}_a)$, on that particular grid level. In a realistic system, the collocation represents most of the computational time spent in the evaluation of the QM/MM electrostatic potential that is around 60%–80%. Afterward, the multi-grid expansion $V_i^{\text{QM/MM}}(\mathbf{r}, \mathbf{R}_a)$ is sequentially interpolated starting from the

coarsest grid level up to the finest level, using real-space interpolator and restrictor operators.

Using the real-space multi-grid operators together with the GEEP expansion, the prefactor in the evaluation of the QM/MM electrostatic potential has been lowered from N_f^3 to N_c^3 , where N_f is the number of grid points on the finest grid and N_c is the number of grid points on the coarsest grid. The computational cost of the other operations for evaluating the electrostatic potential, such as the mapping of the Gaussians and the interpolations, becomes negligible in the limit of a large MM system, usually more than 600–800 MM atoms.

Using the fact that grids are commensurate ($N_f/N_c = 2^{3(N_{\text{grid}}-1)}$), and employing for every calculation four grid levels, the speed-up factor is around 512 (2^9). This means that the present implementation is two orders of magnitude faster than the direct analytical evaluation of the potential on the grid.

2. QM/MM for periodic systems

The effect of the periodic replicas of the MM subsystem is only in the long-range term and comes entirely from the residual function $R_{\text{low}}(\mathbf{r}, \mathbf{R}_a)$,

$$\begin{aligned} V_{\text{recip}}^{\text{QM/MM}}(\mathbf{r}, \mathbf{R}_a) &= \sum_{\mathbf{L}} \sum_a q_a v_a^{\text{recip}} \\ &= \sum_{\mathbf{L}} \sum_a q_a R_{\text{low}}(|\mathbf{r} - \mathbf{R}_a + \mathbf{L}|), \end{aligned} \quad (169)$$

where \mathbf{L} labels the infinite sums over the period replicas. Performing the same manipulation used in the Ewald summation,³⁴⁹ the previous equation can be computed more efficiently in reciprocal space, i.e.,

$$V_{\text{recip}}^{\text{QM/MM}}(\mathbf{r}_i, \mathbf{R}_a) = L^{-3} \sum_{\mathbf{k}} \sum_a q_a \tilde{R}_{\text{low}}(\mathbf{k}) \cos[2\pi \mathbf{k} \cdot (\mathbf{r}_i - \mathbf{R}_a)]. \quad (170)$$

The term $\tilde{R}_{\text{low}}(\mathbf{k})$, representing the Fourier transform of the smooth electrostatic potential, can be analytically evaluated via

$$\begin{aligned} \tilde{R}_{\text{low}}(\mathbf{k}) &= \left[\frac{4\pi}{|\mathbf{k}|^2} \right] \exp\left(-\frac{|\mathbf{k}|^2 r_{c,a}^2}{4}\right) \\ &\quad - \sum_{N_g} A_g(\pi)^{\frac{3}{2}} G_g^3 \exp\left(-\frac{G_g^2 |\mathbf{k}|^2}{4}\right). \end{aligned} \quad (171)$$

The potential in Eq. (170) can be mapped on the coarsest available grid. Once the electrostatic potential of a single MM charge within periodic boundary conditions is derived, the evaluation of the electrostatic potential due to the MM subsystem is easily computed employing the same multi-grid operators (interpolation and restriction) used for isolated systems.

The description of the long-range QM/MM interaction with periodic boundary conditions requires the description of the QM/MM periodic interactions, which plays a significant role if the QM subsystem has a net charge different from zero or a significant dipole moment. Here, we exploit a technique proposed few years ago by Blöchl,³⁵⁰ for decoupling the periodic images and restoring the correct periodicity also for the QM part. A full and comprehensive description of the methods is reported in Ref. 351.

3. Image charge augmented QM/MM

The IC augmented QM/MM model in CP2K has been developed for the simulation of molecules adsorbed at metallic interfaces.³⁵² In the IC-QM/MM scheme, the adsorbates are described by KS-DFT and the metallic substrate is treated at the MM level of theory. The interactions between the QM and MM subsystems are modeled by empirical potentials of, e.g., the Lennard-Jones-type to reproduce dispersion and Pauli repulsion. Polarization effects due to electrostatic screening in metallic conductors are explicitly accounted for by applying the IC formulation.

The charge distribution of the adsorbed molecules generates an electrostatic potential $V_e(\mathbf{r})$, which extends into the substrate. If the electrostatic response of the metal is not taken into account in our QM/MM setup, the electrostatic potential has different values at different points \mathbf{r} inside the metal slab, as illustrated in Fig. 12. However, the correct physical behavior is that $V_e(\mathbf{r})$ induces an electrostatic response such that the potential in the metal is zero or at least constant. Following Ref. 353, we describe the electrostatic response of the metallic conductor by introducing an image charge distribution $\rho_r(\mathbf{r})$, which is modeled by a set of Gaussian charges $\{g_a\}$ centered at the metal atoms so that

$$\rho_m(\mathbf{r}) = \sum_a c_a g_a(\mathbf{r}, \mathbf{R}_a), \quad (172)$$

where \mathbf{R}_a are the positions of the metal atoms. The unknown expansion coefficients c_a are determined self-consistently imposing the constant-potential condition, i.e., the potential $V_m(\mathbf{r})$ generated by $\rho_m(\mathbf{r})$ screens $V_e(\mathbf{r})$ within the metal so that $V_e(\mathbf{r}) + V_m(\mathbf{r}) = V_0$, where V_0 is a constant potential that can be different from zero if an external potential is applied. The modification of the electrostatic potential upon application of the IC correction is shown in Fig. 12. Details on the underlying theory and implementation can be found in Ref. 352.

The IC-QM/MM scheme provides a computationally efficient way to describe the MM-based electrostatic interactions between the adsorbate and the metal in a fully self-consistent fashion, allowing the charge densities of the QM and MM parts to mutually modify each other. Except for the positions of the metal atoms, no input parameters are required. The computational overhead compared to a conventional QM/MM scheme is in the current implementation

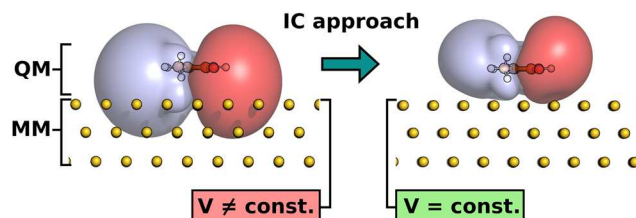


FIG. 12. Simulation of molecules at metallic surfaces using the IC-QM/MM approach. The isosurface of the electrostatic potential at 0.0065 a.u. (red) and -0.0065 a.u. (blue) for a single thymine molecule at Au(111). Left: standard QM/MM approach, where the electrostatic potential of the molecule extends beyond the surface into the metal. Right: IC-QM/MM approach reproducing the correct electrostatics expected for a metallic conductor.

negligible. The IC augmentation adds an attractive interaction because $\rho_m(\mathbf{r})$ mirrors the charge distribution of the molecule but has the opposite sign (see Ref. 352 for the energy expression). Therefore, the IC scheme strengthens the interactions between molecules and metallic substrates, in particular, if adsorbates are polar or even charged. It also partially accounts for the rearrangement of the electronic structure of the molecules when they approach the surface.³⁵²

The IC-QM/MM approach is a good model for systems, where the accurate description of the molecule–molecule interactions is of primary importance and the surface acts merely as a template. For such systems, it is sufficient when the augmented QM/MM model predicts the structure of the first adsorption layer correctly. The IC-QM/MM approach has been applied, for example, to liquid water/Pt(111) interfaces,^{352,354} organic thin-film growth on Au(100),³⁵⁵ as well as aqueous solutions of DNA molecules at gold interfaces.^{356,357} Instructions on how to set up an IC-QM/MM calculation with CP2K can be found in Ref. 358.

4. Partial atomic charges from electrostatic potential fitting

Electrostatic potential (ESP) fitting is a popular approach to determine a set of partial atomic point charges $\{q_a\}$, which can then be used in, e.g., classical MD simulations to model electrostatic contributions in the force-field. The fitting procedure is applied such that the charges $\{q_a\}$ optimally reproduce a given potential V_{QM} obtained from a quantum chemical calculation. To avoid unphysical values for the fitted charges, restraints (and constraints) are often set for q_a , which are then called RESP charges.³⁵⁹

The QM potential is typically obtained from a preceding DFT or Hartree–Fock calculation. The difference between $V_{\text{QM}}(\mathbf{r})$ and the potential $V_{\text{RESP}}(\mathbf{r})$ generated by $\{q_a\}$ is minimized in a least squares fitting procedure at defined grid points \mathbf{r}_k . The residual R_{esp} that is minimized is

$$R_{\text{esp}} = \frac{1}{N} \sum_k (V_{\text{QM}}(\mathbf{r}_k) - V_{\text{RESP}}(\mathbf{r}_k))^2, \quad (173)$$

where N is the total number of grid points included in the fit. The choice of $\{\mathbf{r}_k\}$ is system dependent, and choosing $\{\mathbf{r}_k\}$ carefully is important to obtain meaningful charges. For isolated molecules or porous periodic structures (e.g., metal–organic frameworks), $\{\mathbf{r}_k\}$ are sampled within a given spherical shell around the atoms defined by a minimal and a maximal radius. The minimal radius is usually set to values larger than the van der Waals radius to avoid fitting in spatial regions, where the potential varies rapidly, which would result in a destabilization of the fit. As a general guideline, the charges should be fitted in the spatial regions relevant for interatomic interactions. CP2K offers also the possibility to fit RESP charges for slab-like systems. For these systems, it is important to reproduce the potential accurately above the surface where adsorption processes take place. The sampling technique is flexible enough to follow a corrugation of the surface (see Ref. 360).

When calculating RESP charges for periodic systems, periodic boundary conditions have to be employed for the calculation of V_{RESP} . In CP2K, the RESP point charges q_a are represented as Gaussian functions. The resulting charge distribution is presented on a regular real-space grid and the GPW formalism is employed to

obtain a periodic RESP potential. Details of the implementation can be found in Ref. 360.

CP2K features also a GPW implementation of the REPEAT method,³⁶¹ which is a modification of the RESP fitting for periodic systems. The residual in Eq. (173) is modified such that the variance of the potentials instead of the absolute difference is fitted,

$$R_{\text{repeat}} = \frac{1}{N} \sum_k (V_{\text{QM}}(\mathbf{r}_k) - V_{\text{RESP}}(\mathbf{r}_k) - \delta)^2, \quad (174)$$

where

$$\delta = \frac{1}{N} \sum_k (V_{\text{QM}}(\mathbf{r}_k) - V_{\text{RESP}}(\mathbf{r}_k)). \quad (175)$$

The REPEAT method was originally introduced to account for the arbitrary offset of the electrostatic potential in infinite systems, which depends on the numerical method used. In CP2K, V_{QM} and V_{RESP} are both evaluated with the same method, the GPW approach, and have thus the same offset. However, fitting the variance is an easier task than fitting the absolute difference in the potentials and stabilizes significantly the periodic fit. Using the REPEAT method is thus recommended for the periodic case, in particular for systems that are not slab-like. For the latter, the potential above the surface changes very smoothly and we find that $\delta \approx 0$.³⁶⁰

The periodic RESP and REPEAT implementation in CP2K has been used to obtain atomic charges for surface systems, such as corrugated hexagonal boron nitride (hBN) monolayers on Rh(111).³⁶⁰ These charges were then used in MD QM/MM simulations of liquid water films (QM) on the hBN@Rh(111) substrate (MM). Other applications comprise two-dimensional periodic supramolecular structures,³⁶² metal–organic frameworks,^{363–365} as well as graphene.³⁶⁶ Detailed instructions on how to set up a RESP or REPEAT calculation with CP2K can be found under Ref. 367.

B. Density functional embedding theory

Quantum embedding theories are multi-level approaches applying different electronic structure methods to subsystems, interacting with each other quantum-mechanically.³⁶⁸ Density functional embedding theory (DFET) introduces high-order correlation to a chemically relevant subsystem (cluster), whereas the environment and the interaction between the cluster and the environment are described with DFT via the unique local embedding potential $v_{\text{emb}}(\mathbf{r})$.³⁶⁹ The simplest method to calculate the total energy is in the first-order perturbation theory fashion,

$$E_{\text{total}}^{\text{DFET}} = E_{\text{total}}^{\text{DFT}} + (E_{\text{cluster,emb}}^{\text{CW}} - E_{\text{cluster,emb}}^{\text{DFT}}), \quad (176)$$

where $E_{\text{cluster,emb}}^{\text{DFT}}$ and $E_{\text{env,emb}}^{\text{DFT}}$ are DFT energies of the embedded subsystems, whereas $E_{\text{cluster,emb}}^{\text{CW}}$ is the energy of the embedded cluster at the correlated wavefunction (CW) level of theory. All these entities are computed with an additional one-electron embedding term in the Hamiltonian.

The embedding potential is obtained from the condition that the sum of embedded subsystem densities should reconstruct the DFT density of the total system. This can be achieved by maximizing the Wu–Yang functional with respect to v_{emb} ,³⁷⁰

$$W[V_{\text{emb}}] = E_{\text{cluster}}[\rho_{\text{cluster}}] + E_{\text{env}}[\rho_{\text{env}}] + \int V_{\text{emb}}(\rho_{\text{total}} - \rho_{\text{cluster}} - \rho_{\text{env}}) d\mathbf{r}, \quad (177)$$

with the functional derivative to be identical to the density difference $\frac{\delta W}{\delta V_{\text{emb}}} = \rho_{\text{total}} - \rho_{\text{cluster}} - \rho_{\text{env}}$.

The DFET workflow consists of computing the total system with DFT and obtaining v_{emb} by repeating DFT calculations on the subsystems with updated embedding potential until the total DFT density is reconstructed. When the condition is fulfilled, the higher-level embedded calculation is performed on the cluster. The main computational cost comes from the DFT calculations. The cost is reduced by a few factors such as employing a wavefunction extrapolation from the previous iterations via the time-reversible always stable predictor-corrector integrator.³¹²

The DFET implementation is available for closed and open electronic shells in unrestricted and restricted open-shell variants for the latter. It is restricted to GPW calculations with pseudopotentials describing the core electrons (i.e., full-electron methods are currently not available). Any method implemented within QUICKSTEP is available as a higher-level method, including hybrid DFT, MP2, and RPA. It is possible to perform property calculations on the embedded clusters using an externally provided v_{emb} . The subsystems can employ different basis sets, although they must share the same PW grid.

In our implementation, v_{emb} may be represented on the real-space grid, as well as using a finite Gaussian basis set, the first option being preferable, as it allows a much more accurate total density reconstruction and is computationally cheaper.

C. Implicit solvent techniques

AIMD simulations of solutions, biological systems, or surfaces in the presence of a solvent are often computationally dominated by the calculation of the explicit solvent portion, which may easily amount to 70% of the total number of atoms in a model system. Although the first and second sphere or layer of the solvent molecules around a solute or above a surface might have a direct impact via chemical bonding, for instance, via a hydrogen bonding network, the bulk solvent mostly interacts electrostatically as a continuous dielectric medium. This triggered the development of methods treating the bulk solvent implicitly. Such implicit solvent methods have also the further advantage that they provide a statistically averaged effect of the bulk solvent. That is beneficial for AIMD simulations that consider only a relatively small amount of bulk solvent and quite short sampling times. The self-consistent reaction field (SCRF) in a sphere, which goes back to the early work by Onsager,³⁷¹ is possibly the most simple implicit solvent method implemented in CP2K.^{372,373}

More recent continuum solvation models such as the conductor-like screening model (COSMO),³⁷⁴ as well as the polarizable continuum model (PCM),^{375,376} take also into account the explicit shape of the solute. The solute-solvent interface is defined as the surface of a cavity around the solute. This cavity is constructed by interlocking spheres centered on the atoms or atomic groups composing the solute.³⁷⁷ This introduces a discontinuity of the dielectric function at the solute-solvent interface and thus causes non-continuous atomic forces, which may impact the convergence of structural relaxations, as well as the energy conservation within

AIMD runs. To overcome these problems, Fattberg and Gygi proposed a smoothed self-consistent dielectric model function of the electronic density $\rho^{\text{elec}}(\mathbf{r})$,^{378,379}

$$\epsilon[\rho^{\text{elec}}(\mathbf{r})] = 1 + \frac{\epsilon_0 - 1}{2} \left(1 + \frac{1 - (\rho^{\text{elec}}(\mathbf{r})/\rho_0)^{2\beta}}{1 + (\rho^{\text{elec}}(\mathbf{r})/\rho_0)^{2\beta}} \right), \quad (178)$$

with the model parameters β and ρ_0 , which fulfills the asymptotic behavior

$$\epsilon(\mathbf{r}) \equiv \epsilon[\rho^{\text{elec}}(\mathbf{r})] = \begin{cases} 1, & \text{large } \rho^{\text{elec}}(\mathbf{r}) \\ \epsilon_0, & \rho^{\text{elec}}(\mathbf{r}) \rightarrow 0. \end{cases} \quad (179)$$

The method requires a self-consistent iteration of the polarization charge density spreading across the solute-solvent interface in each SCF iteration step, since the dielectric function depends on the actual electronic density $\rho^{\text{elec}}(\mathbf{r})$. This may cause a non-negligible computational overhead depending on the convergence behavior.

More recently, Andreussi *et al.* proposed in the framework of a revised self-consistent continuum solvation (SCCS) model an improved piecewise defined dielectric model function,³⁸⁰

$$\epsilon[\rho^{\text{elec}}(\mathbf{r})] = \begin{cases} 1, & \rho^{\text{elec}}(\mathbf{r}) > \rho_{\text{max}} \\ \exp(t), & \rho_{\text{min}} \leq \rho^{\text{elec}}(\mathbf{r}) \leq \rho_{\text{max}} \\ \epsilon_0, & \rho^{\text{elec}}(\mathbf{r}) < \rho_{\text{min}}, \end{cases} \quad (180)$$

with $t = t[\ln(\rho^{\text{elec}}(\mathbf{r}))]$ that employs a more elaborated switching function for the transition from the solute to the solvent region using the smooth function

$$t(x) = \frac{\ln \epsilon_0}{2\pi} \left[2\pi \frac{(\ln \rho_{\text{max}} - x)}{(\ln \rho_{\text{max}} - \ln \rho_{\text{min}})} - \sin \left(2\pi \frac{(\ln \rho_{\text{max}} - x)}{(\ln \rho_{\text{max}} - \ln \rho_{\text{min}})} \right) \right]. \quad (181)$$

Both models are implemented in CP2K.^{381,382}

The solvation free energy ΔG^{sol} can be computed by

$$\Delta G^{\text{sol}} = \Delta G^{\text{el}} + G^{\text{rep}} + G^{\text{dis}} + G^{\text{cav}} + G^{\text{tm}} + P\Delta V \quad (182)$$

as the sum of the electrostatic contribution $\Delta G^{\text{el}} = G^{\text{el}} - G^0$, where G^0 is the energy of the solute in vacuum, the repulsion energy $G^{\text{rep}} = \alpha S$, the dispersion energy $G^{\text{dis}} = \beta V$, and the cavitation energy $G^{\text{cav}} = \gamma S$, with adjustable solvent specific parameters α , β , and γ .³⁸³ Therein, S and V are the (quantum) surface and volume of the solute cavity, respectively, which are evaluated in CP2K based on the quantum surface

$$S = \int \left[\vartheta \left(\rho^{\text{elec}}(\mathbf{r}) - \frac{\Delta}{2} \right) - \left(\rho^{\text{elec}}(\mathbf{r}) + \frac{\Delta}{2} \right) \right] \frac{|\nabla \rho^{\text{elec}}(\mathbf{r})|}{\Delta} d\mathbf{r} \quad (183)$$

and quantum volume

$$V = \int \vartheta[\rho^{\text{elec}}(\mathbf{r})] d\mathbf{r}, \quad (184)$$

a definition that was introduced by Cococcioni *et al.* with

$$\vartheta[\rho^{\text{elec}}(\mathbf{r})] = \frac{\epsilon_0 - \epsilon[\rho^{\text{elec}}(\mathbf{r})]}{\epsilon_0 - 1} \quad (185)$$

using the smoothed dielectric function either from Eq. (179) or Eq. (180).^{380,384} The thermal motion term G^{tm} and the volume change term $P\Delta V$ are often ignored and are not yet available in CP2K.

D. Poisson solvers

The Poisson equation describes how the electrostatic potential $V(\mathbf{r})$, as a contribution to the Hamiltonian, relates to the charge distribution within the system with electron density $n(\mathbf{r})$. In a general form, which incorporates a non-homogeneous position-dependent dielectric constant $\epsilon(\mathbf{r})$, the equation is formulated as

$$-\nabla \cdot (\epsilon(\mathbf{r}) \nabla V(\mathbf{r})) = 4\pi n(\mathbf{r}), \quad (186)$$

endowed with some suitable boundary conditions determined by the physical characteristics or setup of the considered system.

Under the assumption that everywhere inside the simulation domain, the dielectric constant is equal to that of free space, i.e., 1, and for periodically repeated systems or boundary conditions for which the analytical Green's function of the standard Poisson operator is known, such as free, wire, or surface boundary conditions, a variety of methods have been proposed to solve Eq. (186). Among these approaches, CP2K implements the conventional PW scheme, Ewald summation based techniques,^{349,385–388} the Martyna–Tuckerman method,¹² and a wavelet based Poisson solver.^{14,15} Within the context of continuum solvation models,³⁷⁵ where solvent effects are included implicitly in the form of a dielectric medium,³⁷⁸ the code also provides the solver proposed by Andreussi *et al.* that can solve Eq. (186) subject to periodic boundary conditions and with ϵ defined as a function of the electronic density.³⁸⁰

Despite the success of these methods in numerous scenarios (for some recent applications, see, e.g., Refs. 381 and 389), the growing interest in simulating atomistic systems with more complex boundary configurations, e.g., nanoelectronic devices, at an *ab initio* level of theory, demands for Poisson solvers with capabilities that exceed those of the existing solvers. To this end, a generalized Poisson solver has been developed and implemented in CP2K with the following key features:³⁹⁰

- The solver converges exponentially with respect to the density cutoff.
- Periodic or homogeneous von Neumann (zero normal electric field) boundary conditions can be imposed on the boundaries of the simulation cell. The latter models insulating interfaces such as air-semiconductor and oxide-semiconductor interfaces in a transistor.
- Fixed electrostatic potentials (Dirichlet-type boundary conditions) can be enforced at arbitrarily shaped regions within the domain. These correspond, e.g., to source, drain, and gate contacts of a nanoscale device.
- The dielectric constant can be expressed as any sufficiently smooth function.

Therefore, the solver offers advantages associated with the two categories of Poisson solvers, i.e., PW and real-space-based methods.

The imposition of the above-mentioned boundary setups is accomplished by solving an equivalent constrained variational problem that reads as

$$\text{Find } (V, \lambda) \text{ s.t. } J(V, \lambda) = \min_u \max_\mu J(u, \mu), \quad (187)$$

where

$$J(u, \mu) = \frac{1}{2} \int_{\Omega} \epsilon |\nabla u|^2 d\mathbf{r} - \int_{\Omega} 4\pi n u d\mathbf{r} + \int_{\Omega_D} \mu (u - V_D) d\mathbf{r}. \quad (188)$$

Here, V_D is the potential applied at the predefined subdomain Ω_D that may have an arbitrary geometry, whereas $u(V)$ satisfies the desired boundary conditions at the boundaries of the cell Ω and $\mu(\lambda)$ are Lagrange multipliers introduced to enforce the constant potential. Note that the formulation of the problem within Eqs. (187) and (188) is independent of how ϵ is defined.

Furthermore, consistent ionic forces have been derived and implemented that make the solver applicable to a wider range of applications, such as energy-conserving BOMD³⁹¹ as well as Ehrenfest dynamics simulations.³⁹²

XIII. DBCSR LIBRARY

DBCSR has been specifically designed to efficiently perform block-sparse and dense matrix operations on distributed multicore CPU and GPU systems, covering a range of occupancy from 0.01% up to dense.^{114,393,394} The library is written in Fortran and is freely available under the GPL license from <https://github.com/cp2k/dbcsr>. Operations include sum, dot product, and multiplication of matrices, and the most important operations on single matrices, such as transpose and trace.

The DBCSR library was developed to unify the previous disparate implementations of distributed dense and sparse matrix data structures and the operations among them. The chief performance optimization target is parallel matrix multiplication. Its performance objectives have been comparable to ScaLAPACK's PDGEMM for dense or nearly dense matrices²⁰¹ while achieving high performance when multiplying sparse matrices.¹¹⁴

DBCSR matrices are stored in a blocked compressed sparse row (CSR) format distributed over a two-dimensional grid of P MPI processes. Although the library accepts single and double precision complex and real numbers, it is only optimized for the double precision real type. The sizes of the blocks in the data structure are not arbitrary but are determined by the properties of the data stored in the matrix, such as the basis sets of the atoms in the molecular system. While this property keeps the blocks in the matrices dense, it often results in block sizes that are suboptimal for computer processing, e.g., blocks of 5×5 , 5×13 , and 13×13 for the H and O atoms described by a DZVP basis set.

To achieve high performance for the distributed multiplication of two possibly sparse matrices, several techniques have been used. One is to focus on reducing communication costs among MPI ranks. Another is to optimize the performance of multiplying communicated data on a node by introducing a separation of concerns: *what* will be multiplied vs *performing* the multiplications. The CPUs on a node determine what needs to be multiplied, creating batches of work. These batches are handled by hardware-specific drivers, such as CPUs, GPUs, or a combination of both. DBCSR's GPU backend, LIBSMM_ACC, supports both NVIDIA and AMD GPUs via

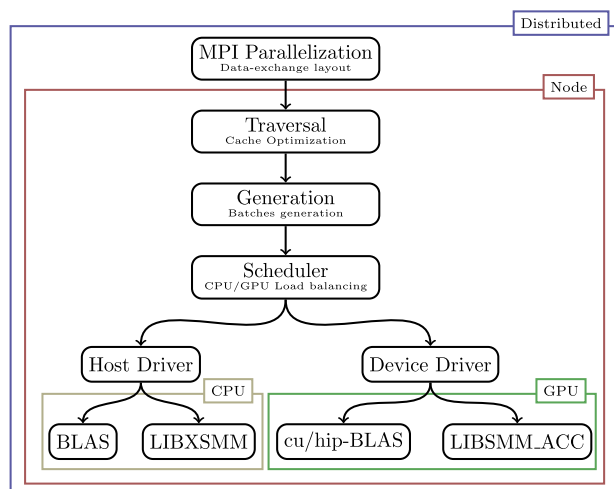


FIG. 13. Schema of the DBCSR library for the matrix–matrix multiplication (see text for description).

CUDA and HIP, respectively. A schema of the library is shown in Fig. 13.

A. Message passing interface parallelization

At the top level, we have the MPI parallelization. The data-layout exchange is implemented with two different algorithms, depending on the sizes of the involved matrices in the multiplications:

- for general matrices (any size), we use the Cannon algorithm, where the amount of communicated data by each process scales as $\mathcal{O}(1/\sqrt{P})$,^{114,395} and
- only for “tall-and-skinny” matrices (one large dimension), we use an optimized algorithm, where the amount of communicated data by each process scales as $\mathcal{O}(1)$.³⁹⁶

The communications are implemented with asynchronous point-to-point MPI calls. The local multiplication will start as soon as all the data have arrived at the destination process, and it is possible to overlap the local computation with the communication if the network allows that.

B. Local multiplication

The local computation consists of pairwise multiplications of small dense matrix blocks, with dimensions $(m \times k)$ for A blocks and $(k \times n)$ for B blocks. It employs a cache-oblivious matrix traversal to fix the order in which matrix blocks need to be computed, to improve memory locality (Traversal phase in Fig. 13). First, the algorithm loops over A matrix row-blocks and then, for each row-block, over B matrix column-blocks. Then, the corresponding multiplications are organized in batches (Generation phase in Fig. 13), where each batch consists of maximum 30 000 multiplications. During the Scheduler phase, a static assignment of batches with a given A matrix row-block to OpenMP threads is employed to avoid data-race conditions. Finally, batches assigned to each thread

can be computed in parallel on the CPU and/or executed on a GPU. For the GPU execution, batches are organized in such a way that the transfers between the host and the GPU are minimized. The multiplication kernels take full advantage of the opportunities for coalesced memory operations and asynchronous operations. Moreover, a double-buffering technique, based on CUDA streams and events, is used to maximize the occupancy of the GPU and to hide the data transfer latency.³⁹³ When the GPU is fully loaded, the computation may be simultaneously done on the CPU. Multi-GPU execution on the same node is made possible by distributing the cards to multiple MPI ranks via a round-robin assignment.

C. Batched execution

Processing batches of small-matrix-multiplications (SMMs) have to be highly efficient. For this reason, specific libraries were developed that outperform vendor BLAS libraries, namely, LIBSMM_ACC (previously called LIBCUSMM, which is part of DBCSR) for GPUs,³⁹⁶ as well as LIBXSMM for CPUs.^{397,398}

In LIBSMM_ACC, GPU kernels are just-in-time (JIT) compiled at runtime. This allows us to reduce DBCSR’s compile time by more than half and its library’s size by a factor of 6 compared to generating and compiling kernels ahead-of-time. Figure 14 illustrates the performance gain that can be observed since the introduction of the JIT framework: because including a new (m,n,k) -kernel to the library incurs no additional compile time, nor does it bloat the library size, all available (m,n,k) batched-multiplications can be run on GPUs, leading to important speed-ups.

LIBSMM_ACC’s GPU kernels are parametrized over seven parameters, affecting the memory usages and patterns of the multiplication algorithms, the amount of work and number of threads per CUDA block, the number of matrix elements computed by each

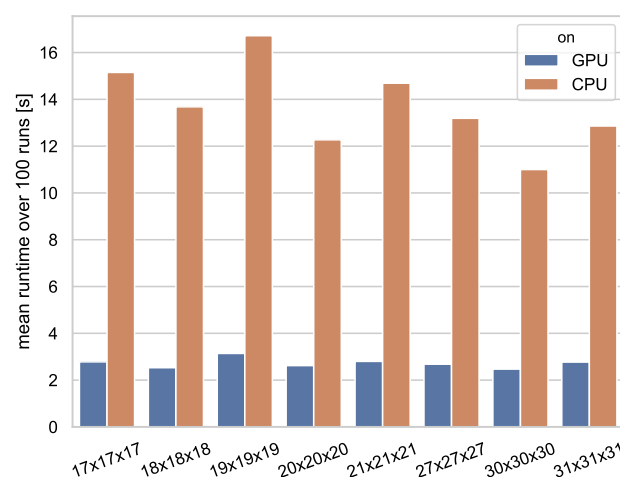


FIG. 14. Comparison of DBCSR dense multiplication of square matrices of size 10 000, dominated by different sub-matrix block sizes. With the JIT framework in place, these blocks are batch-multiplied on a GPU using LIBSMM_ACC, instead of on the CPU using LIBXSMM. Multiplications were run on a heterogeneous Piz Daint CRAY XC50 node containing a 12 core Intel Haswell CPU and on a NVIDIA V100 GPU.

CUDA thread, and the tiling sizes. An autotuning framework finds the optimal parameter set for each (m,n,k)-kernel, exploring about 100 000 possible parameter combinations. These parameter combinations result in vastly different performances: for example, the batched multiplication of kernel $5 \times 6 \times 6$ can vary between 10 Gflops and 276 Gflops in performance, with only 2% of the possible parameter combinations yielding performances within 10% of the optimum. Kernels are autotuned for NVIDIA's P100 [1649 (m,n,k)s] and V100 [1235 (m,n,k)s], as well as for AMD's Mi50 (10 kernels). These autotuning data are then used to train a machine learning performance prediction model, which finds near-optimal parameter sets for the rest of the 75 000 available (m,n,k)-kernels. In this way, the library can achieve a speed-up in the range of 2–4× with respect to batched DGEMM in cuBLAS for $\{m, n, k\} < 32$, while the effect becomes less prominent for larger sizes.³⁹⁸ The performance of LIBSMM_ACC and LIBXSMM saturates for $\{m, n, k\} > 80$, for which DBCSR directly calls cuBLAS or BLAS, respectively.

LIBXSMM generates kernels using a machine model, e.g., considering the size of the (vector-)register file. Runtime code generation in LIBXSMM does not depend on the compiler or flags used and can naturally exploit instruction set extensions presented by CPUID features. The raw machine code is emitted JIT with no compilation phase and one-time cost (10k–50k kernels per second depending on the system and kernel). The generated code is kept for the runtime of an application (such as normal functions) and ready-for-call at a rate of 20M–40M dispatches per second. Both timings include the cost of thread-safety or potentially concurrent requests. Any such overhead is even lower when amortized with homogeneous batches (same kernel on a per batch basis).

D. Outlook

An arithmetic intensity of $AI = 0.3\text{--}3$ FLOPS/Byte (double-precision) is incurred by CP2K's needs. This low AI is bound by the STREAM Triad vector kernel, given that $C = C + A * B$ is based on SMMs rather than scalar values.³⁹⁷ Since memory bandwidth is precious, lower or mixed precision (including single-precision) can be of interest given emerging support in CPUs and GPUs (see Sec. XV B).

XIV. INTERFACES TO OTHER PROGRAMS

Beside containing native f77 and f90 interfaces, CP2K also provides a library, which can be used in your own executable,^{399,400} and comes with a small helper program called CP2K-shell that includes a simple interactive command line interface with a well defined, parsable syntax. In addition, CP2K can be interfaced with external programs such as i-PI,⁴⁰¹ PLUMED,⁴⁰² or PyRETIS,⁴⁰³ and its AIMD trajectories analyzed using TAMk in Ref. 404, MD-Tracks,⁴⁰⁵ and TRAVIS,⁴⁰⁶ to name just a few.

A. Non-equilibrium Green's function formalism

The non-equilibrium Green's function (NEGF) formalism provides a comprehensive framework for studying the transport mechanism across open nanoscale systems in the ballistic as well as scattering regimes.^{407–409} Within the NEGF formalism, all

observables of a system can be obtained from the single-particle Green's functions. In the ballistic regime, the main quantity of interest is the retarded Green's function G , which solves the following equation:

$$(E \cdot S - H - \Sigma) \cdot G(E) = I, \quad (189)$$

where E is the energy level of the injected electron; S and H are the overlap and Hamiltonian matrices, respectively; I is the identity matrix; and Σ is the boundary (retarded) self-energy that incorporates the coupling between the central active region of the device and the semi-infinite leads. A major advantage of the NEGF formalism is that no restrictions are made on the choice of the Hamiltonian that means empirical, semi-empirical, or *ab initio* Hamiltonians can be used in Eq. (189). Utilizing a typically Kohn–Sham, DFT Hamiltonian, the DFT + NEGF method has established itself as a powerful technique due to its capability of modeling charge transport through complex nanostructures without any need for system-specific parameterizations.^{410,411} With the goal of designing a DFT + NEGF simulator that can handle systems of unprecedented size, e.g., composed of tens of thousands of atoms, the quantum transport solver OMEN has been integrated within CP2K.^{412–414} For a given device configuration, CP2K constructs the DFT Hamiltonian and overlap matrices. The matrices are passed to OMEN where the Hamiltonian is modified for open boundaries and charge and current densities are calculated in the NEGF or, equivalently, wave function/quantum transmitting boundary formalisms.⁴¹⁵ Due to the robust and highly efficient algorithms implemented in OMEN and exploiting hybrid computational resources effectively, devices with realistic sizes and structural configurations can now be routinely simulated.^{416–418}

B. SIRIUS: Plane wave density functional theory support

CP2K supports computations of the electronic ground state, including forces and stresses,^{419,420} in a PW basis. The implementation relies on the quantum engine SIRIUS.⁴²¹

The SIRIUS library has full support for GPUs and CPUs and brings additional functionalities to CP2K. Collinear and non-collinear magnetic systems with or without spin–orbit coupling can be studied with both pseudopotential PW⁴²² and full-potential linearized augmented PW methods.^{423,424} All GGA XC functionals implemented in *libxc* are available.⁴²⁵ Hubbard corrections are based on Refs. 426 and 427.

To demonstrate a consistent implementation of the PW energies, forces, and stresses in CP2K + SIRIUS, an AIMD of a Si₇Ge supercell has been performed in the isothermal–isobaric NPT ensemble. To establish the accuracy of the calculations, the ground state energy of the obtained atomic configurations at each time step has been recomputed with QUANTUM ESPRESSO (QE) using the same cutoff parameters, XC functionals, and pseudopotentials.^{428,429} In Fig. 15, it is shown that the total energy is conserved up to 10^{-6} hartree. It also shows the evolution of the potential energies calculated with CP2K + SIRIUS (blue line) and QE (red dashed line). The two curves are shifted by a constant offset, which can be attributed to differences in various numerical schemes employed. After removal of this constant difference, the agreement between the two codes is of the order of 10^{-6} hartree.

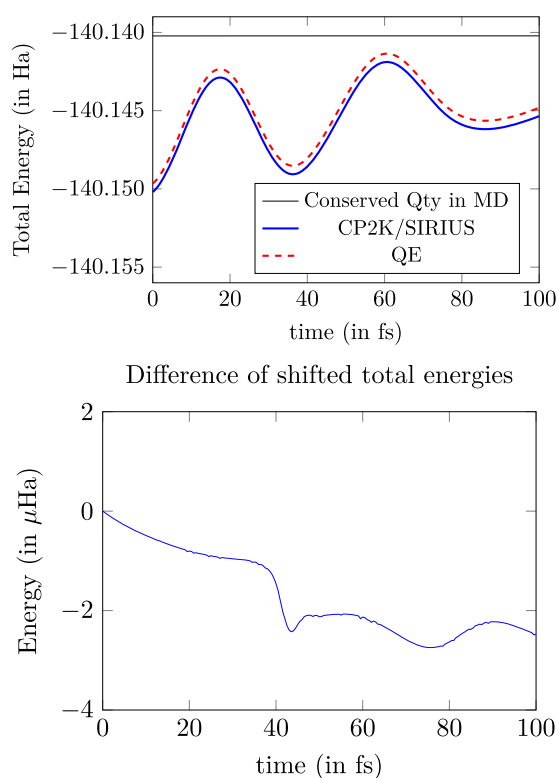


FIG. 15. Comparison of potential energies between CP2K + SIRIUS and QE during an AIMD simulation. The total energy (black curve) remains constant over time (it is a conserved quantity in the NPT ensemble), while the potential energy calculated with CP2K + SIRIUS (blue curve) and QE (red dashed curve) have the same time evolution up to a constant shift in energy (see the lower panel).

XV. TECHNICAL AND COMMUNITY ASPECTS

With an average growth of 200 lines of code per day, the CP2K code comprises currently more than a million lines of code. Most of CP2K is written in Fortran95, with elements from Fortran03 and extensions such as OpenMP, OpenCL, and CUDA C. It also employs various external libraries in order to incorporate new features and to decrease the complexity of CP2K, thereby increasing the efficiency and robustness of the code. The libraries range from basic functionality such as MPI,⁴³⁰ fast Fourier transforms (FFTW),⁴³¹ dense linear algebra (BLAS, LAPACK, ScaLAPACK, and ELPA),^{201–203} to more specialized chemical libraries to evaluate electron repulsion integrals (libint) and XC functionals (libxc).^{425,432} CP2K itself can be built as a library, allowing for easy access to some part of the functionality by external programs. Having lean, library-like interfaces within CP2K has facilitated the implementation of features such as farming (running various inputs within a single job), general input parameter optimization, path-integral MD, or Gibbs-ensemble MC.

Good performance and massive parallel scalability are key features of CP2K. This is achieved using a multi-layer structure of specifically designed parallel algorithms. On the highest level, parallel algorithms are based on message passing with MPI, which is

suitable for distributed memory architectures, augmented with shared memory parallelism based on threading and programmed using OpenMP directives. Ongoing work aims at porting the main algorithms of CP2K to accelerators and GPUs as these energy efficient devices become more standard in supercomputers. At the lowest level, auto-generated and auto-tuned code allows for generating CPU-specific libraries that deliver good performance without a need for dedicated code development.

A. Hardware acceleration

CP2K provides mature hardware accelerator support for GPUs and emerging support for FPGAs. The most computationally intensive kernels in CP2K are handled by the DBCSR library (see Sec. XIII), which decomposes sparse matrix operations into BLAS and LIBXSMM library calls. For GPUs, DBCSR provides a CUDA driver, which offloads these functions to corresponding libraries for GPUs (LIBXSMM_ACC and cuBLAS). FFT operations can also be offloaded to GPUs using the cuFFT library. For FPGAs, however, there exist currently no general and widely used FFT libraries that could be reused. Hence, a dedicated FFT interface has been added to CP2K and an initial version of an FFT library for offloading complex-valued, single-precision 3D FFTs of sizes 32^3 – 128^3 to Intel Arria 10 and Stratix 10 FPGAs has been released.⁴³³

B. Approximate computing

The AC paradigm is an emerging approach to devise techniques for relaxing the exactness to improve the performance and efficiency of the computation.⁴³⁴ The most common method of numerical approximation is the use of adequate data-widths in computationally intensive application kernels. Although many scientific applications use double-precision floating-point by default, this accuracy is not always required. Instead, low-precision arithmetic and mixed-precision arithmetic have been very effective for the computation of inverse matrix roots²²³ or solving systems of linear equations.^{435–438} Driven by the growing popularity of artificial neural networks that can be evaluated and trained with reduced precision, hardware accelerators have gained improved low-precision computing support. For example, NVidia V100 GPU achieves 7 GFlops in double-precision and 15 GFlops in single-precision but up to 125 GFlops tensor performance with half-precision floating point operations. Using low-precision arithmetic is thus essential for exploiting the performance potential of upcoming hardware accelerators.

However, in scientific computing, where the exactness of all computed results is of paramount importance, attenuating accuracy requirements is not an option. Yet, for specific problem classes, it is possible to bound or compensate the error introduced by inaccurate arithmetic. In CP2K, we can apply the AC paradigm to the force computation within MD and rigorously compensate the numerical inaccuracies due to low-accuracy arithmetic operations and still obtain exact ensemble-averaged expectation values, as obtained by time averages of a properly modified Langevin equation.

For that purpose, the notion of the second-generation CPMD method is reversed and we model the nuclear forces as

$$\mathbf{F}_I^N = \mathbf{F}_I + \Xi_I^N, \quad (190)$$

where Ξ_I^N is an additive white noise that is due to a low-precision force computation on a GPU or FPGA-based accelerator. Given that Ξ_I^N is unbiased, i.e.,

$$\langle \mathbf{F}_I(0) \Xi_I^N(t) \rangle \cong 0 \quad (191)$$

holds, it is nevertheless possible to accurately sample the Boltzmann distribution by means of a modified Langevin-type equation,^{225,439,440}

$$M_I \ddot{\mathbf{R}}_I = \mathbf{F}_I^N - \gamma_N M_I \dot{\mathbf{R}}_I. \quad (192)$$

This is to say that the noise, as originating from a low-precision computation, can be thought of as an additive white noise channel associated with a hitherto unknown damping coefficient γ_N , which satisfies the fluctuation–dissipation theorem

$$\langle \Xi_I^N(0) \Xi_I^N(t) \rangle \cong 2\gamma_N M_I k_B T \delta(t). \quad (193)$$

As before, the specific value of γ_N is determined in such a way so as to generate the correct average temperature, as measured by the equipartition theorem of Eq. (140), by means of the adaptive Langevin technique of Jones and co-workers.^{316–318}

C. Benchmarking

To demonstrate the parallel scalability of the various DFT-based and post-Hartree-Fock electronic structure methods implemented in CP2K, strong-scaling plots with respect to the number of CPU-cores are shown in Fig. 16.

The benefit of the AC paradigm, described in Sec. XV B, in terms of the reduction in wall time for the computation of the

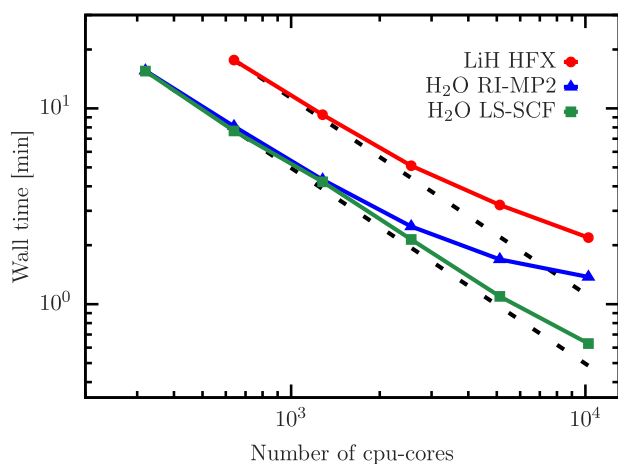


FIG. 16. Parallel scaling of CP2K calculations up to thousands of CPU-cores: single-point energy calculation using GAPW with exact Hartree-Fock exchange for a periodic 216 atom lithium hydride crystal (red line with circles), single-point energy calculation using RI-MP2 for 64 water molecules with periodic boundary conditions (blue line with triangles), and a single-point energy calculation in linear-scaling DFT for 2048 water molecules in the condensed phase using a DZVP MOLOPT basis set (green line with boxes). All calculations have been performed with up to 256 nodes (10 240 CPU-cores) of the Noctua system at PC².

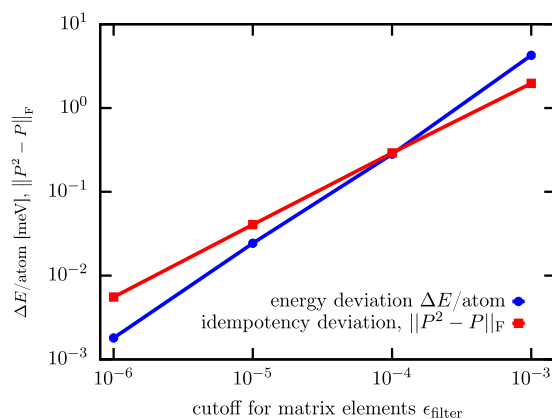


FIG. 17. Energy deviation and deviation from the idempotency condition of the density matrix as a function of matrix truncation threshold for the same system and calculation as in Fig. 6. For the reference energy, a calculation with $\epsilon_{\text{filter}} = 10^{-7}$ was used.

STMV virus, which contains more than one million atoms, is illustrated in Fig. 6. Using the periodic implementation in CP2K of the GFN-xTB model,²³⁶ an increase in efficiency of up to one order of magnitude can be observed within the most relevant matrix-sqrt and matrix-sign operations of the sign-method described in Sec. VIII E. The corresponding violation of the idempotency condition of the density matrix and the energy deviation are shown in Fig. 17. The resulting error within the nuclear forces can be assumed to be white and hence can be compensated by means of the modified Langevin equation of Eq. (192).

D. MOLOPT basis set and delta test

Even though CP2K supports different types of Gaussian-type basis sets, the MOLOPT type basis set has been found to perform particularly well for a large number of systems and also targets a wide range of chemical environments, including the gas phase, interfaces, and the condensed phase.²⁶ These generally rather contracted basis sets, which include diffuse primitives, are obtained by minimizing a linear combination of the total energy and the condition number of the overlap matrix for a set of molecules with respect to the exponents and contraction coefficients of the full basis. To verify reproducibility of DFT codes in the condensed matter community, the Δ -gauge (194)

$$\Delta_i(a, b) = \sqrt{\frac{\int_{0.94V_{0,i}}^{1.06V_{0,i}} (E_{b,i}(V) - E_{a,i}(V))^2 dV}{0.12V_{0,i}}} \quad (194)$$

based on the Birch–Murnaghan equation of state has been established.⁴⁴¹ To gauge our Δ -test values, we have compared them to the ones obtained by the plane wave code Abinit,⁴⁴² where exactly the same dual-space Goedecker–Teter–Hutter (GTH) pseudopotentials as described in Sec. II B have been employed.^{23–25} As shown in Fig. 18, CP2K generally performs fairly well, with the remaining

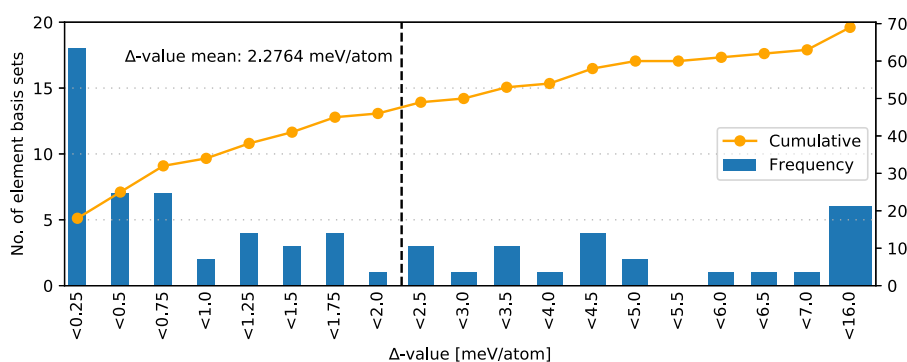


FIG. 18. Δ -values for DFT calculations (using the PBE XC functional and GTH pseudopotentials) of first to fifth row elements without Co and Lu, as computed between the best-performing MOLOPT [i.e., DZVP, TZVP, or TZV2P(X)] basis sets and WIEN2k reference results.⁴⁴³ For comparison, the corresponding average for Abinit is 2.1 meV/atom for the semicore and 6.3 meV/atom for the regular pseudopotentials, respectively.⁴⁴⁴

deviation being attributed to the particular pseudodiagonalization approach.

E. CP2K workflows

With the abundance of computing power, computational scientists are tackling ever more challenging problems. As a consequence, today's simulations often require complex workflows. For example, first, the initial structure is prepared, and then, a series of geometry optimizations at increasing levels of theory are performed, before the actual observables can be computed, which then finally need to be post-processed and analyzed. There is a strong desire to automate these workflows, which not only saves time but also makes them reproducible and shareable. CP2K is interfaced with two popular frameworks for automation of such workflows: The Atomic Simulation Environment (ASE)⁴⁴⁵ and the Automated Interactive Infrastructure and Database for Computational Science (AiiDA).⁴⁴⁶

The ASE framework is a Python library that is built around the *Atoms* and the *Calculator* classes. An *Atoms* object stores the properties of individual atoms such as their positions and atomic numbers. A *Calculator* object provides a simple unified API to a chemistry code such as CP2K. Calculation is performed by passing an *Atoms* object to a *Calculator* object, which then returns energies, nuclear forces, and other observables. For example, running a CP2K calculation of a single water molecule requires just a few lines of Python code:

Based on these two powerful primitives, the ASE provides a rich library of common building blocks including structure

```
$ export ASE_CP2K_COMMAND = "cp2kshell.sopt"
$ python
>>> from ase.calculators.cp2k import CP2K
>>> from ase.build import molecule
>>> calc = CP2K()
>>> atoms = molecule('H2O', calculator=calc)
>>> atoms.center(vacuum=2.0)
>>> print(atoms.get_potential_energy())
-467.191035845
```

generation, MD, local and global geometry optimizations, transition-state methods, vibration analysis, and many more. As such, the ASE is an ideal tool for quick prototyping and automatizing a small number of calculations.

The AiiDA framework, however, aims to enable the emerging field of high-throughput computations.⁴⁴⁷ Within this approach, databases of candidate materials are automatically screened for the desired target properties. Since a project typically requires thousands of computations, very robust workflows are needed that can also handle rare failure modes gracefully. To this end, the AiiDA framework provides a sophisticated event-based workflow engine. Each workflow step is a functional building block with well defined inputs and outputs. This design allows the AiiDA engine to trace the data dependencies throughout the workflows and thereby record the provenance graph of every result in its database.

F. GitHub and general tooling

The CP2K source code is publicly available and hosted in a Git repository on <https://github.com/cp2k/cp2k>. While we still maintain a master repository similar to the previous Subversion repository, the current development process via *pull requests* fosters code reviews and discussion, while the integration with a custom *continuous integration* system based on the Google Cloud Platform ensures the stability of the *master branch* by mandating successful regression tests prior to merging the code change.⁴⁴⁸ The *pull request* based testing is augmented with a large number of additional regression testers running on different supercomputers,⁴⁴⁹ providing over 80% code coverage across all variants (MPI, OpenMP, CUDA/HIP, and FPGA) of CP2K.⁴⁵⁰ These tests are run periodically, rather than triggered live by Git *commits*, to work around limitations imposed by the different sites and developers are being informed automatically in the case of test failures.

Over time, additional code analysis tools have been developed to help avoid common pitfalls and maintain consistent code style over the large code base of over 1×10^6 lines of code. Some of them—like our *fprettify* tool⁴⁵¹—have now been adopted by other Fortran-based codes.⁴⁵² To simplify the workflow of the developers, all code analysis tools not requiring compilation are now invoked automatically on each Git *commit* if the developer has setup the *pre-commit* hooks for her *clone* of the CP2K Git repository.⁴⁵³ Since CP2K has a significant number of optional dependencies, a series

of *toolchain* scripts have been developed to facilitate the installation, currently providing the reference environment for running the regression tests. Alternatively, CP2K packages are directly available for the Linux distributions Debian,⁴⁵⁴ Fedora,⁴⁵⁵ as well as Arch Linux,⁴⁵⁶ thanks to the efforts of M. Banck, D. Mierzejewski, and A. Kudelin. In addition, direct CP2K support is provided by the HPC package management tools Spack and EasyBuild,^{457,458} thanks to M. Culpo and K. Hoste.

ACKNOWLEDGMENTS

T.D.K. received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 716142). J.V. was supported by an ERC Starting Grant (No. 277910), V.V.R. was supported by the Swiss National Science Foundation in the form of Ambizione Grant (No. PZ00P2_174227), and R.Z.K. was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through Discovery Grants (RGPIN-2016-0505). T.D.K. and C.P. kindly acknowledges funding from Paderborn University's research award for "GreenIT". C.P. and M.L. received funding from the German Research Foundation (DFG) under the project PerficienCC (grant agreement No PL 595/2-1). G.K.S. and C.J.M. were supported by the Department of Energy's Basic Energy Sciences, Chemical Sciences, Geosciences, and Biosciences Division. UK based work was funded under the embedded CSE programme of the ARCHER UK National Supercomputing Service (<http://www.archer.ac.uk>), Grant Nos. eCSE03-011, eCSE06-6, eCSE08-9, and eCSE13-17, and the EPSRC (EP/P022235/1) grant "Surface and Interface Toolkit for the Materials Chemistry Community." The project received funding via the CoE MaX as part of the Horizon 2020 program (Grant No. 824143), the Swiss Platform For Advanced Scientific Computing (PASC), and the National Centre of Competence in Research (NCCR) MARVEL. Computational resources were provided by the Swiss National Supercomputing Centre (CSCS) and Compute Canada. The generous allocation of computing time on the FPGA-based supercomputer "Noctua" at PC² is kindly acknowledged.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

- ¹D. Marx and J. Hutter, *Ab Initio Molecular Dynamics: Basic Theory and Advanced Methods* (Cambridge University Press, 2009).
- ²J. Hutter, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **2**, 604 (2011).
- ³T. D. Kühne, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **4**, 391 (2014).
- ⁴M. Krack and M. Parrinello, "Quickstep: Make the atoms dance," in *High Performance Computing in Chemistry*, NIC Series Vol. 25, edited by J. Grotendorst (NIC-Directors, 2004), pp. 29–51.
- ⁵J. VandeVondele, M. Krack, F. Mohamed, M. Parrinello, T. Chassaing, and J. Hutter, *Comput. Phys. Commun.* **167**, 103 (2005).
- ⁶B. G. Lippert, J. Hutter, and M. Parrinello, *Mol. Phys.* **92**, 477 (1997).
- ⁷W. Kohn and L. J. Sham, *Phys. Rev.* **140**, A1133 (1965).
- ⁸C. Roetti, "The crystal code," in *Quantum-Mechanical Ab-Initio Calculation of the Properties of Crystalline Materials*, edited by C. Pisani (Springer Berlin Heidelberg, 1996), Vol. 67, pp. 125–138.
- ⁹K. N. Kudin and G. E. Scuseria, *Chem. Phys. Lett.* **289**, 611 (1998).
- ¹⁰D. Marx and J. Hutter, in *Modern Methods and Algorithms of Quantum Chemistry*, edited by J. Grotendorst (John von Neumann Institute for Computing, Forschungszentrum Jülich, Jülich, 2000), pp. 301–449 (first edition, paperback, ISBN 3–00–005618–1) or pp. 329–477 (second edition, hardcover, ISBN 3–00–005834–6), see <http://www.theochem.rub.de/go/cprev.html>.
- ¹¹J. W. Eastwood and D. R. K. Brownrigg, *J. Comput. Phys.* **32**, 24 (1979).
- ¹²G. J. Martyna and M. E. Tuckerman, *J. Chem. Phys.* **110**, 2810 (1999).
- ¹³P. Mináry, M. E. Tuckerman, K. A. Pihakari, and G. J. Martyna, *J. Chem. Phys.* **116**, 5351 (2002).
- ¹⁴L. Genovese, T. Deutsch, A. Neelov, S. Goedecker, and G. Beylkin, *J. Chem. Phys.* **125**, 074105 (2006).
- ¹⁵L. Genovese, T. Deutsch, and S. Goedecker, *J. Chem. Phys.* **127**, 054704 (2007).
- ¹⁶M. Dion, H. Rydberg, E. Schröder, D. C. Langreth, and B. I. Lundqvist, *Phys. Rev. Lett.* **92**, 246401 (2004).
- ¹⁷G. Román-Pérez and J. M. Soler, *Phys. Rev. Lett.* **103**, 096102 (2009).
- ¹⁸P. Pulay, *Mol. Phys.* **17**, 197 (1969).
- ¹⁹A. Dal Corso and R. Resta, *Phys. Rev. B* **50**, 4327 (1994).
- ²⁰L. C. Balbás, J. L. Martins, and J. M. Soler, *Phys. Rev. B* **64**, 165110 (2001).
- ²¹J. Schmidt, J. VandeVondele, I.-F. W. Kuo, D. Sebastiani, J. I. Siepmann, J. Hutter, and C. J. Mundy, *J. Phys. Chem. B* **113**, 11959 (2009).
- ²²P. E. Blöchl, *Phys. Rev. B* **50**, 17953 (1994).
- ²³C. Hartwigsen, S. Goedecker, and J. Hutter, *Phys. Rev. B* **58**, 3641 (1998).
- ²⁴S. Goedecker, M. Teter, and J. Hutter, *Phys. Rev. B* **54**, 1703 (1996).
- ²⁵M. Krack, *Theor. Chem. Acc.* **114**, 145 (2005).
- ²⁶J. VandeVondele and J. Hutter, *J. Chem. Phys.* **127**, 114105 (2007).
- ²⁷E. J. Baerends, D. E. Ellis, and P. Ros, *Chem. Phys.* **2**, 41 (1973).
- ²⁸D. Golze, M. Iannuzzi, and J. Hutter, *J. Chem. Theory Comput.* **13**, 2202 (2017).
- ²⁹<https://www.cp2k.org/howto:ic-qmmm>, 2017.
- ³⁰D. Golze, N. Benedikter, M. Iannuzzi, J. Wilhelm, and J. Hutter, *J. Chem. Phys.* **146**, 034105 (2017).
- ³¹G. Lippert, J. Hutter, and M. Parrinello, *Theor. Chem. Acc.* **103**, 124 (1999).
- ³²M. Krack and M. Parrinello, *Phys. Chem. Chem. Phys.* **2**, 2105 (2000).
- ³³M. Krack, A. Gambirasio, and M. Parrinello, *J. Chem. Phys.* **117**, 9409 (2002).
- ³⁴L. Triguero, L. G. M. Pettersson, and H. Ågren, *Phys. Rev. B* **58**, 8097 (1998).
- ³⁵M. Iannuzzi and J. Hutter, *Phys. Chem. Chem. Phys.* **9**, 1599 (2007).
- ³⁶M. Iannuzzi, *J. Chem. Phys.* **128**, 204506 (2008).
- ³⁷D. J. Adams, S. Chappellet, F. Lincker, M. Ibn-Elhaj, B. Watts, M. Iannuzzi, D. Šišak Jung, C. A. Pignedoli, and D. Passerone, *J. Phys. Chem. C* **118**, 15422 (2014).
- ³⁸L. Weinhardt, E. Ertan, M. Iannuzzi, M. Weigand, O. Fuchs, M. Bär, M. Blum, J. D. Denlinger, W. Yang, E. Umbach, M. Odelius, and C. Heske, *Phys. Chem. Chem. Phys.* **17**, 27145 (2015).
- ³⁹P. Müller, K. Karhan, M. Krack, U. Gerstmann, W. G. Schmidt, M. Bauer, and T. D. Kühne, *J. Comput. Chem.* **40**, 712 (2019).
- ⁴⁰V. da Cruz, F. Gel'mukhanov, S. Eckert, M. Iannuzzi, E. Ertan, A. Pietzsch, R. Couto, J. Niskanen, M. Fondell, M. Dantz, T. Schmitt, X. Lu, D. McNally, R. Jay, V. Kimberg, A. Föhlisch, and M. Odelius, *Nat. Commun.* **10**, 1013 (2019).
- ⁴¹R. Declerck, E. Pauwels, V. Van Speybroeck, and M. Waroquier, *Phys. Rev. B* **74**, 245103 (2006).
- ⁴²J. Schmidt, J. Hutter, H.-W. Spiess, and D. Sebastiani, *Chem. Phys. Chem.* **9**, 2313 (2008).
- ⁴³V. Weber, M. Iannuzzi, S. Giani, J. Hutter, R. Declerck, and M. Waroquier, *J. Chem. Phys.* **131**, 014106 (2009).
- ⁴⁴A. Mondal, M. W. Gaultois, A. J. Pell, M. Iannuzzi, C. P. Grey, J. Hutter, and M. Kaupp, *J. Chem. Theory Comput.* **14**, 377 (2018).
- ⁴⁵R. O. Jones and O. Gunnarsson, *Rev. Mod. Phys.* **61**, 689 (1989).
- ⁴⁶R. O. Jones, *Rev. Mod. Phys.* **87**, 897 (2015).
- ⁴⁷W. Kohn, *Rev. Mod. Phys.* **71**, 1253 (1999).
- ⁴⁸J. A. Pople, *Rev. Mod. Phys.* **71**, 1267 (1999).
- ⁴⁹A. D. Becke, *Phys. Rev. A* **38**, 3098 (1988).
- ⁵⁰C. Lee, W. Yang, and R. G. Parr, *Phys. Rev. B* **37**, 785 (1988).

- ⁵¹M. Guidon, F. Schiffmann, J. Hutter, and J. VandeVondele, *J. Chem. Phys.* **128**, 214104 (2008).
- ⁵²M. Guidon, J. Hutter, and J. VandeVondele, *J. Chem. Theory Comput.* **5**, 3010 (2009).
- ⁵³M. Guidon, J. Hutter, and J. VandeVondele, *J. Chem. Theory Comput.* **6**, 2348 (2010).
- ⁵⁴C. Adriaanse, J. Cheng, V. Chau, M. Sulpizi, J. VandeVondele, and M. Sprik, *J. Phys. Chem. Lett.* **3**, 3411 (2012).
- ⁵⁵J. Cheng, X. Liu, J. VandeVondele, M. Sulpizi, and M. Sprik, *Acc. Chem. Res.* **47**, 3522 (2014).
- ⁵⁶J. Heyd, G. E. Scuseria, and M. Ernzerhof, *J. Chem. Phys.* **118**, 8207 (2003).
- ⁵⁷J. Heyd, G. E. Scuseria, and M. Ernzerhof, *J. Chem. Phys.* **124**, 219906 (2006).
- ⁵⁸J. Spencer and A. Alavi, *Phys. Rev. B* **77**, 193110 (2008).
- ⁵⁹J. Paier, C. V. Diaconu, G. E. Scuseria, M. Guidon, J. VandeVondele, and J. Hutter, *Phys. Rev. B* **80**, 174114 (2009).
- ⁶⁰T. Ohto, M. Dodia, J. Xu, S. Imoto, F. Tang, F. Zysk, T. D. Kühne, Y. Shigeta, M. Bonn, X. Wu, and Y. Nagata, *J. Phys. Chem. Lett.* **10**, 4914 (2019).
- ⁶¹C. Möller and M. S. Plesset, *Phys. Rev.* **46**, 618 (1934).
- ⁶²T. Helgaker, P. Jørgensen, and J. Olsen, *Molecular Electronic-Structure Theory* (John Wiley & Sons, Ltd., Chichester, 2000).
- ⁶³S. Grimme, *J. Chem. Phys.* **124**, 034108 (2006).
- ⁶⁴M. Feyereisen, G. Fitzgerald, and A. Komornicki, *Chem. Phys. Lett.* **208**, 359 (1993).
- ⁶⁵F. Weigend and M. Häser, *Theor. Chem. Acc.* **97**, 331 (1997).
- ⁶⁶Y. Jung, R. C. Lochan, A. D. Dutoi, and M. Head-Gordon, *J. Chem. Phys.* **121**, 9793 (2004).
- ⁶⁷A. Takatsuka, S. Ten-no, and W. Hackbusch, *J. Chem. Phys.* **129**, 044112 (2008).
- ⁶⁸M. Del Ben, J. Hutter, and J. VandeVondele, *J. Chem. Theory Comput.* **8**, 4177 (2012).
- ⁶⁹M. Del Ben, J. Hutter, and J. VandeVondele, *J. Chem. Theory Comput.* **9**, 2654 (2013).
- ⁷⁰M. Del Ben, J. Hutter, and J. VandeVondele, *J. Chem. Phys.* **143**, 102803 (2015).
- ⁷¹V. V. Rybkin and J. VandeVondele, *J. Chem. Theory Comput.* **12**, 2214 (2016).
- ⁷²M. Del Ben, M. Schönherr, J. Hutter, and J. VandeVondele, *J. Phys. Chem. Lett.* **4**, 3753 (2013).
- ⁷³M. Del Ben, M. Schönherr, J. Hutter, and J. VandeVondele, *J. Phys. Chem. Lett.* **5**, 3066 (2014).
- ⁷⁴M. Del Ben, J. Hutter, and J. VandeVondele, *J. Chem. Phys.* **143**, 054506 (2015).
- ⁷⁵M. Del Ben, J. VandeVondele, and B. Slater, *J. Phys. Chem. Lett.* **5**, 4122 (2014).
- ⁷⁶J. Wilhelm, J. VandeVondele, and V. V. Rybkin, *Angew. Chem., Int. Ed.* **58**, 3890 (2019).
- ⁷⁷J. P. Perdew and K. Schmidt, *AIP Conf. Proc.* **577**, 1 (2001).
- ⁷⁸F. Furche, *Phys. Rev. B* **64**, 195120 (2001).
- ⁷⁹M. Fuchs and X. Gonze, *Phys. Rev. B* **65**, 235109 (2002).
- ⁸⁰T. Miyake, F. Aryasetiawan, T. Kotani, M. van Schilfgaarde, M. Usuda, and K. Terakura, *Phys. Rev. B* **66**, 245103 (2002).
- ⁸¹F. Furche and T. Van Voorhis, *J. Chem. Phys.* **122**, 164106 (2005).
- ⁸²M. Fuchs, Y.-M. Niquet, X. Gonze, and K. Burke, *J. Chem. Phys.* **122**, 094116 (2005).
- ⁸³J. Harl and G. Kresse, *Phys. Rev. B* **77**, 045136 (2008).
- ⁸⁴D. Lu, Y. Li, D. Rocca, and G. Galli, *Phys. Rev. Lett.* **102**, 206411 (2009).
- ⁸⁵A. Heßelmann, *Phys. Rev. A* **85**, 012517 (2012).
- ⁸⁶J. Toulouse, I. C. Gerber, G. Jansen, A. Savin, and J. G. Ángyán, *Phys. Rev. Lett.* **102**, 096404 (2009).
- ⁸⁷J. Harl and G. Kresse, *Phys. Rev. Lett.* **103**, 056401 (2009).
- ⁸⁸X. Ren, P. Rinke, C. Joas, and M. Scheffler, *J. Mater. Sci.* **47**, 7447 (2012).
- ⁸⁹A. Heßelmann and A. Görling, *Mol. Phys.* **109**, 2473 (2011).
- ⁹⁰A. Heßelmann and A. Görling, *Mol. Phys.* **108**, 359 (2010).
- ⁹¹J. Paier, B. G. Janesko, T. M. Henderson, G. E. Scuseria, A. Grüneis, and G. Kresse, *J. Chem. Phys.* **132**, 094103 (2010).
- ⁹²A. Grüneis, M. Marsman, J. Harl, L. Schimka, and G. Kresse, *J. Chem. Phys.* **131**, 154115 (2009).
- ⁹³F. Furche, *J. Chem. Phys.* **129**, 114105 (2008).
- ⁹⁴D. Rocca, *J. Chem. Phys.* **140**, 18A501 (2014).
- ⁹⁵B. G. Janesko, T. M. Henderson, and G. E. Scuseria, *J. Chem. Phys.* **130**, 081105 (2009).
- ⁹⁶B. Mussard, D. Rocca, G. Jansen, and J. G. Ángyán, *J. Chem. Theory Comput.* **12**, 2191 (2016).
- ⁹⁷H. Eshuis, J. Yarkony, and F. Furche, *J. Chem. Phys.* **132**, 234114 (2010).
- ⁹⁸M. Del Ben, O. Schütt, T. Wentz, P. Messmer, J. Hutter, and J. VandeVondele, *Comput. Phys. Commun.* **187**, 120 (2015).
- ⁹⁹M. Kaltak, J. Klimeš, and G. Kresse, *J. Chem. Theory Comput.* **10**, 2498 (2014).
- ¹⁰⁰F. Aryasetiawan, T. Miyake, and K. Terakura, *Phys. Rev. Lett.* **88**, 166401 (2002).
- ¹⁰¹H.-V. Nguyen and S. de Gironcoli, *Phys. Rev. B* **79**, 205114 (2009).
- ¹⁰²H. Eshuis, J. E. Bates, and F. Furche, *Theor. Chem. Acc.* **131**, 1084 (2012).
- ¹⁰³X. Ren, P. Rinke, V. Blum, J. Wierferink, A. Tkatchenko, A. Sanfilippo, K. Reuter, and M. Scheffler, *New J. Phys.* **14**, 053020 (2012).
- ¹⁰⁴P. García-González, J. J. Fernández, A. Marini, and A. Rubio, *J. Phys. Chem. A* **111**, 12458 (2007).
- ¹⁰⁵J. Harl, L. Schimka, and G. Kresse, *Phys. Rev. B* **81**, 115126 (2010).
- ¹⁰⁶B. Xiao, J. Sun, A. Ruzsinszky, J. Feng, and J. P. Perdew, *Phys. Rev. B* **86**, 094109 (2012).
- ¹⁰⁷M. Rohlfing and T. Bredow, *Phys. Rev. Lett.* **101**, 266106 (2008).
- ¹⁰⁸X. Ren, P. Rinke, and M. Scheffler, *Phys. Rev. B* **80**, 045402 (2009).
- ¹⁰⁹A. Marini, P. García-González, and A. Rubio, *Phys. Rev. Lett.* **96**, 136404 (2006).
- ¹¹⁰F. Mittendorfer, A. Garhofer, J. Redinger, J. Klimeš, J. Harl, and G. Kresse, *Phys. Rev. B* **84**, 201401 (2011).
- ¹¹¹Y. Li, D. Lu, H.-V. Nguyen, and G. Galli, *J. Phys. Chem. A* **114**, 1944 (2010).
- ¹¹²H. F. Schurkus and C. Ochsenfeld, *J. Chem. Phys.* **144**, 031101 (2016).
- ¹¹³J. Wilhelm, P. Seewald, M. Del Ben, and J. Hutter, *J. Chem. Theory Comput.* **12**, 5851 (2016).
- ¹¹⁴U. Borštnik, J. VandeVondele, V. Weber, and J. Hutter, *Parallel Comput.* **40**, 47 (2014).
- ¹¹⁵J. VandeVondele, U. Borštnik, and J. Hutter, *J. Chem. Theory Comput.* **8**, 3565 (2012).
- ¹¹⁶J. Demmel, D. Eliahu, A. Fox, S. Kamil, B. Lipshitz, O. Schwartz, and O. Spillinger, in *2013 IEEE 27th International Symposium on Parallel and Distributed Processing* (IEEE, 2013), pp. 261–272.
- ¹¹⁷L. Hedin, *Phys. Rev.* **139**, A796 (1965).
- ¹¹⁸M. S. Hybertsen and S. G. Louie, *Phys. Rev. B* **34**, 5390 (1986).
- ¹¹⁹F. Aryasetiawan and O. Gunnarsson, *Rep. Prog. Phys.* **61**, 237 (1998).
- ¹²⁰D. Golze, M. Dvorak, and P. Rinke, *Front. Chem.* **7**, 377 (2019).
- ¹²¹B. Kippelen and J.-L. Brédas, *Energy Environ. Sci.* **2**, 251 (2009).
- ¹²²E. J. Baerends, O. V. Gritsenko, and R. van Meer, *Phys. Chem. Chem. Phys.* **15**, 16408 (2013).
- ¹²³M. S. Hybertsen and S. G. Louie, *Phys. Rev. Lett.* **55**, 1418 (1985).
- ¹²⁴L. Gallandi, N. Marom, P. Rinke, and T. Körzdörfer, *J. Chem. Theory Comput.* **12**, 605 (2016).
- ¹²⁵J. W. Knight, X. Wang, L. Gallandi, O. Dolgounitcheva, X. Ren, J. V. Ortiz, P. Rinke, T. Körzdörfer, and N. Marom, *J. Chem. Theory Comput.* **12**, 615 (2016).
- ¹²⁶L. Gallandi and T. Körzdörfer, *J. Chem. Theory Comput.* **11**, 5391 (2015).
- ¹²⁷J. F. Janak, *Phys. Rev. B* **18**, 7165 (1978).
- ¹²⁸X. Blase, C. Attaccalite, and V. Olevano, *Phys. Rev. B* **83**, 115103 (2011).
- ¹²⁹S.-H. Ke, *Phys. Rev. B* **84**, 205415 (2011).
- ¹³⁰J. Wilhelm, M. Del Ben, and J. Hutter, *J. Chem. Theory Comput.* **12**, 3623 (2016).
- ¹³¹J. Wilhelm and J. Hutter, *Phys. Rev. B* **95**, 235123 (2017).

- ¹³²Z. Yi, Y. Ma, Y. Zheng, Y. Duan, and H. Li, *Adv. Mater. Interfaces* **6**, 1801175 (2018).
- ¹³³H. N. Rojas, R. W. Godby, and R. J. Needs, *Phys. Rev. Lett.* **74**, 1827 (1995).
- ¹³⁴H. J. Vidberg and J. W. Serene, *J. Low Temp. Phys.* **29**, 179 (1977).
- ¹³⁵D. Golze, J. Wilhelm, M. J. van Setten, and P. Rinke, *J. Chem. Theory Comput.* **14**, 4856 (2018).
- ¹³⁶D. Golze, L. Keller, and P. Rinke, *J. Phys. Chem. Lett.* **11**, 1840 (2020).
- ¹³⁷J. Wilhelm, D. Golze, L. Talirz, J. Hutter, and C. A. Pignedoli, *J. Phys. Chem. Lett.* **9**, 306 (2018).
- ¹³⁸M. van Schilfgaarde, T. Kotani, and S. Faleev, *Phys. Rev. Lett.* **96**, 226402 (2006).
- ¹³⁹F. Bruneval, N. Vast, and L. Reining, *Phys. Rev. B* **74**, 045102 (2006).
- ¹⁴⁰N. Marom, F. Caruso, X. Ren, O. T. Hofmann, T. Körzdörfer, J. R. Chelikowsky, A. Rubio, M. Scheffler, and P. Rinke, *Phys. Rev. B* **86**, 245127 (2012).
- ¹⁴¹D. Beyer, S. Wang, C. A. Pignedoli, J. Melidone, B. Yuan, C. Li, J. Wilhelm, P. Ruffieux, R. Berger, K. Müllen, R. Fasel, and X. Feng, *J. Am. Chem. Soc.* **141**, 2843 (2019); J. I. Urgel, S. Mishra, H. Hayashi, J. Wilhelm, C. A. Pignedoli, M. Di Giovannantonio, R. Widmer, M. Yamashita, N. Hieda, P. Ruffieux, H. Yamada, and R. Fasel, *Nat. Commun.* **10**, 861 (2019); K. Xu, J. I. Urgel, K. Eimre, M. Di Giovannantonio, A. Keerthi, H. Komber, S. Wang, A. Narita, R. Berger, P. Ruffieux, C. A. Pignedoli, J. Liu, K. Müllen, R. Fasel, and X. Feng, *J. Am. Chem. Soc.* **141**, 7726 (2019); X.-Y. Wang, J. I. Urgel, G. B. Barin, K. Eimre, M. Di Giovannantonio, A. Milani, M. Tommasini, C. A. Pignedoli, P. Ruffieux, X. Feng, R. Fasel, K. Müllen, and A. Narita, *ibid.* **140**, 9104 (2018); M. Di Giovannantonio, J. I. Urgel, U. Beser, A. V. Yakutovich, J. Wilhelm, C. A. Pignedoli, P. Ruffieux, A. Narita, K. Müllen, and R. Fasel, *ibid.* **140**, 3532 (2018).
- ¹⁴²J. B. Neaton, M. S. Hybertsen, and S. G. Louie, *Phys. Rev. Lett.* **97**, 216405 (2006).
- ¹⁴³B. Kaduk, T. Kowalczyk, and T. Van Voorhis, *Chem. Rev.* **112**, 321 (2012).
- ¹⁴⁴Q. Wu and T. Van Voorhis, *Phys. Rev. A* **72**, 024502 (2005).
- ¹⁴⁵Q. Wu and T. Van Voorhis, *J. Chem. Phys.* **125**, 164105 (2006).
- ¹⁴⁶N. Holmberg and K. Laasonen, *J. Chem. Theory Comput.* **13**, 587 (2017).
- ¹⁴⁷N. Holmberg and K. Laasonen, *J. Chem. Phys.* **149**, 104702 (2018).
- ¹⁴⁸Q. Wu, C.-L. Cheng, and T. Van Voorhis, *J. Chem. Phys.* **127**, 164119 (2007).
- ¹⁴⁹S. Baroni, P. Giannozzi, and A. Testa, *Phys. Rev. Lett.* **58**, 1861 (1987).
- ¹⁵⁰S. Baroni, S. de Gironcoli, A. Dal Corso, and P. Giannozzi, *Rev. Mod. Phys.* **73**, 515 (2001).
- ¹⁵¹X. Gonze, *Phys. Rev. A* **52**, 1096 (1995).
- ¹⁵²A. Putrino, D. Sebastiani, and M. Parrinello, *J. Chem. Phys.* **113**, 7102 (2000).
- ¹⁵³R. D. King-Smith and D. Vanderbilt, *Phys. Rev. B* **47**, 1651 (1993).
- ¹⁵⁴D. Vanderbilt and R. D. King-Smith, *Phys. Rev. B* **48**, 4442 (1993).
- ¹⁵⁵R. Resta, *Rev. Mod. Phys.* **66**, 899 (1994).
- ¹⁵⁶R. Resta, *Phys. Rev. Lett.* **80**, 1800 (1998).
- ¹⁵⁷N. A. Spaldin, *J. Solid State Chem.* **195**, 2 (2012).
- ¹⁵⁸A. Putrino and M. Parrinello, *Phys. Rev. Lett.* **88**, 1764011 (2002).
- ¹⁵⁹P. Partovi-Azar and T. D. Kühne, *J. Comput. Chem.* **36**, 2188 (2015).
- ¹⁶⁰P. Partovi-Azar, T. D. Kühne, and P. Kaghazchi, *Phys. Chem. Chem. Phys.* **17**, 22009 (2015).
- ¹⁶¹M. Thomas, M. Brehm, R. Fligg, P. Vöhringer, and B. Kirchner, *Phys. Chem. Chem. Phys.* **15**, 6608 (2013).
- ¹⁶²S. Lubert, M. Iannuzzi, and J. Hutter, *J. Chem. Phys.* **141**, 094503 (2014).
- ¹⁶³G. Schreckenbach and T. Ziegler, *Int. J. Quantum Chem.* **61**, 899 (1997).
- ¹⁶⁴A. V. Y.-D. Deyne, E. Pauwels, V. V. Speybroeck, and M. Waroquier, *Phys. Chem. Chem. Phys.* **14**, 10690 (2012).
- ¹⁶⁵N. Marzari and D. Vanderbilt, *Phys. Rev. B* **56**, 12847 (1997).
- ¹⁶⁶G. Berghold, C. J. Mundy, A. H. Romero, J. Hutter, and M. Parrinello, *Phys. Rev. B* **61**, 10040 (2000).
- ¹⁶⁷N. Marzari, A. A. Mostofi, J. R. Yates, I. Souza, and D. Vanderbilt, *Rev. Mod. Phys.* **84**, 1419 (2012).
- ¹⁶⁸D. Sebastiani and M. Parrinello, *J. Phys. Chem. A* **105**, 1951 (2001).
- ¹⁶⁹D. Sebastiani, *Mod. Phys. Lett. B* **17**, 1301 (2003).
- ¹⁷⁰D. Sebastiani, *Chem. Phys. Chem.* **7**, 164 (2006).
- ¹⁷¹D. Sebastiani, G. Goward, I. Schnell, and M. Parrinello, *Comput. Phys. Commun.* **147**, 707 (2002).
- ¹⁷²T. A. Keith and R. F. W. Bader, *Chem. Phys. Lett.* **194**, 1 (1992).
- ¹⁷³T. A. Keith and R. F. W. Bader, *Chem. Phys. Lett.* **210**, 223 (1993).
- ¹⁷⁴M.-L. Bonnet, M. Iannuzzi, D. Sebastiani, and J. Hutter, *J. Phys. Chem. C* **116**, 18577 (2012).
- ¹⁷⁵A. Hoefling, D. T. Nguyen, P. Partovi-Azar, D. Sebastiani, P. Theato, S.-W. Song, and Y. J. Lee, *Chem. Mater.* **30**, 2915 (2018).
- ¹⁷⁶R. Declerck, E. Pauwels, V. Van Speybroeck, and M. Waroquier, *J. Phys. Chem. B* **112**, 1508 (2008).
- ¹⁷⁷E. Pauwels, R. Declerck, T. Verstraelen, B. De Sterck, C. W. M. Kay, V. Van Speybroeck, and M. Waroquier, *J. Phys. Chem. B* **114**, 16655 (2010).
- ¹⁷⁸H. Elgabarty, P. Schmieder, and D. Sebastiani, *Chem. Sci.* **4**, 755 (2013).
- ¹⁷⁹C. Allolio, N. Salas-Illanes, Y. S. Desmukh, M. R. Hansen, and D. Sebastiani, *J. Phys. Chem. B* **117**, 9939 (2013).
- ¹⁸⁰H. Elgabarty, R. Z. Khaliullin, and T. D. Kühne, *Nat. Commun.* **6**, 8318 (2015).
- ¹⁸¹G. Bekcioglu-Neff, C. Allolio, Y. Desmukh, M. Hansen, and D. Sebastiani, *Chem. Phys. Chem.* **17**, 1166 (2016).
- ¹⁸²S. A. Pylaeva, H. Elgabarty, D. Sebastiani, and P. M. Tolstoy, *Phys. Chem. Chem. Phys.* **19**, 26107 (2017).
- ¹⁸³S. Pylaeva, K. L. Ivanov, M. Baldus, D. Sebastiani, and H. Elgabarty, *J. Phys. Chem. Lett.* **8**, 2137 (2017).
- ¹⁸⁴E. Pauwels, J. Asher, M. Kaupp, and M. Waroquier, *Phys. Chem. Chem. Phys.* **13**, 18638 (2011).
- ¹⁸⁵M. E. Casida, in *Recent Advances in Density Functional Methods*, edited by D. P. Chong (World Scientific, Singapore, 1995), Chap. 5, pp. 155–192.
- ¹⁸⁶J. Strand, S. K. Chulkov, M. B. Watkins, and A. L. Shluger, *J. Chem. Phys.* **150**, 044702 (2019).
- ¹⁸⁷A. L. Fetter and J. D. Walecka, *Quantum Theory of Many-Particle Systems*, International Series in Pure and Applied Physics (McGraw-Hill, New York, 1971), Chap. 15, p. 601.
- ¹⁸⁸J. Hutter, *J. Chem. Phys.* **118**, 3928 (2003).
- ¹⁸⁹S. Hirata and M. Head-Gordon, *Chem. Phys. Lett.* **314**, 291 (1999).
- ¹⁹⁰A. Dreuw, J. L. Weisman, and M. Head-Gordon, *J. Chem. Phys.* **119**, 2943 (2003).
- ¹⁹¹M. Iannuzzi, T. Chassaing, T. Wallman, and J. Hutter, *CHIMIA Int. J. Chem.* **59**, 499 (2005).
- ¹⁹²M. Crouzeix, B. Philippe, and M. Sadkane, *SIAM J. Sci. Comput.* **15**, 62 (1994).
- ¹⁹³E. Poli, J. D. Elliott, S. K. Chulkov, M. B. Watkins, and G. Teobaldi, *Front. Chem.* **7**, 210 (2019).
- ¹⁹⁴T. Kunert and R. Schmidt, *Eur. Phys. J. D* **25**, 15 (2003).
- ¹⁹⁵S. Andermatt, J. Cha, F. Schiffrmann, and J. VandeVondele, *J. Chem. Theory Comput.* **12**, 3214 (2016).
- ¹⁹⁶R. G. Gordon and Y. S. Kim, *J. Chem. Phys.* **56**, 3122 (1972).
- ¹⁹⁷Y. S. Kim and R. G. Gordon, *J. Chem. Phys.* **60**, 1842 (1974).
- ¹⁹⁸J. P. Stewart, P. Császár, and P. Pulay, *J. Comput. Chem.* **3**, 227 (1982).
- ¹⁹⁹J. VandeVondele and J. Hutter, *J. Chem. Phys.* **118**, 4365 (2003).
- ²⁰⁰R. McWeeny, *Rev. Mod. Phys.* **32**, 335 (1960).
- ²⁰¹J. Choi, J. Demmel, I. Dhillon, J. Dongarra, S. Ostrouchov, A. Petit, K. Stanley, D. Walker, and R. C. Whaley, *Comput. Phys. Commun.* **97**, 1 (1996).
- ²⁰²T. Auckenthaler, V. Blum, H.-J. Bungartz, T. Huckle, R. Johanni, L. Krämer, B. Lang, H. Lederer, and P. R. Willems, *Parallel Comput.* **37**, 783 (2011).
- ²⁰³A. Marek, V. Blum, R. Johanni, V. Havu, B. Lang, T. Auckenthaler, A. Heinecke, H.-J. Bungartz, and H. Lederer, *J. Phys.: Condens. Matter* **26**, 213201 (2014).
- ²⁰⁴P. O. Löwdin, *J. Chem. Phys.* **18**, 365 (1950).
- ²⁰⁵P. Pulay, *J. Comput. Chem.* **3**, 556 (1982).
- ²⁰⁶P. Pulay, *Chem. Phys. Lett.* **73**, 393 (1980).
- ²⁰⁷R. McWeeny and C. A. Coulson, *Philos. Trans. R. Soc. London, Ser. A* **235**, 496 (1956).
- ²⁰⁸R. Fletcher, *Mol. Phys.* **19**, 55 (1970).

- ²⁰⁹R. Seeger and J. A. Pople, *J. Chem. Phys.* **65**, 265 (1976).
- ²¹⁰I. Stich, R. Car, M. Parrinello, and S. Baroni, *Phys. Rev. B* **39**, 4997 (1989).
- ²¹¹C. Yang, J. C. Meza, and L.-W. Wang, *SIAM J. Sci. Comput.* **29**, 1854 (2007).
- ²¹²V. Weber, J. VandeVondele, J. Hutter, and A. M. N. Niklasson, *J. Chem. Phys.* **128**, 084113 (2008).
- ²¹³N. J. Higham, *Functions of Matrices: Theory and Computation (Other Titles in Applied Mathematics)* (Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008).
- ²¹⁴A. M. N. Niklasson, *Phys. Rev. B* **70**, 193102 (2004).
- ²¹⁵E. Polak and G. Ribiere, *ESAIM: Math. Modell. Numer. Anal.* **3**, 35 (1969).
- ²¹⁶J. Nocedal and S. J. Wright, *Numerical Optimization* (Springer, New York, NY, USA, 2006).
- ²¹⁷J. Kiefer, *Proc. Am. Math. Soc.* **4**, 502 (1953).
- ²¹⁸H.-r. Fang and Y. Saad, *Numer. Linear Algebra Appl.* **16**, 197 (2009).
- ²¹⁹J. M. Martinez, *J. Comput. Appl. Math.* **124**, 97 (2000), part of special issue: Numerical Analysis 2000. Vol. IV: Optimization and Nonlinear Equations.
- ²²⁰K. Baarman and J. VandeVondele, *J. Chem. Phys.* **134**, 244104 (2011).
- ²²¹F. Schiffmann and J. VandeVondele, *J. Chem. Phys.* **142**, 244117 (2015).
- ²²²M. Lass, S. Mohr, H. Wiebeler, T. D. Kühne, and C. Plessl, in *Proceedings of the Platform for Advanced Scientific Computing Conference, PASC'18* (Association for Computing Machinery, New York, NY, USA, 2018).
- ²²³M. Lass, T. D. Kühne, and C. Plessl, *IEEE Embedded Syst. Lett.* **10**, 33 (2018).
- ²²⁴D. Richters, M. Lass, A. Walther, C. Plessl, and T. D. Kühne, *Commun. Comput. Phys.* **25**, 564 (2019).
- ²²⁵D. Richters and T. D. Kühne, *J. Chem. Phys.* **140**, 134109 (2014).
- ²²⁶A. M. N. Niklasson, C. J. Tymczak, and M. Challacombe, *J. Chem. Phys.* **118**, 8611 (2003).
- ²²⁷E. H. Rubensson and A. M. N. Niklasson, *SIAM J. Sci. Comput.* **36**, B147 (2014).
- ²²⁸L. Lin, J. Lu, L. Ying, R. Car, and W. E, *Commun. Math. Sci.* **7**, 755 (2009).
- ²²⁹L. Lin, M. Chen, C. Yang, and L. He, *J. Phys.: Condens. Matter* **25**, 295501 (2013).
- ²³⁰M. Jacquelin, L. Lin, and C. Yang, *ACM Trans. Math. Software* **43**, 1 (2016).
- ²³¹M. Jacquelin, L. Lin, and C. Yang, *Parallel Comput.* **74**, 84 (2018).
- ²³²N. J. Higham, *Numer. Algorithms* **15**, 227 (1997).
- ²³³M. Ceriotti, T. D. Kühne, and M. Parrinello, *J. Chem. Phys.* **129**, 024707 (2008).
- ²³⁴M. Ceriotti, T. D. Kühne, and M. Parrinello, *AIP Conf. Proc.* **1148**, 658 (2009).
- ²³⁵C. Kenney and A. J. Laub, *SIAM J. Matrix Anal. Appl.* **12**, 273 (1991).
- ²³⁶C. Bannwarth, S. Ehlert, and S. Grimme, *J. Chem. Theory Comput.* **15**, 1652 (2019).
- ²³⁷S. F. Boys, *Rev. Mod. Phys.* **32**, 296 (1960).
- ²³⁸J. Pipek and P. G. Mezey, *J. Chem. Phys.* **90**, 4916 (1989).
- ²³⁹Z. Luo and R. Z. Khaliullin, "Direct unconstrained variable-metric localization of one-electron orbitals," *J. Chem. Theory Comput.* (published online, 2020).
- ²⁴⁰O. Matsuoka, *J. Chem. Phys.* **66**, 1245 (1977).
- ²⁴¹H. Stoll and H. Preuß, *Theor. Chem. Acc.* **46**, 11 (1977).
- ²⁴²E. L. Mehler, *J. Chem. Phys.* **67**, 2728 (1977).
- ²⁴³H. Stoll, G. Wagenblast, and H. Preubeta, *Theor. Chim. Acta* **57**, 169 (1980).
- ²⁴⁴P. Ordejón, D. A. Drabold, R. M. Martin, and M. P. Grumbach, *Phys. Rev. B* **51**, 1456 (1995).
- ²⁴⁵C.-K. Skylaris, P. D. Haynes, A. A. Mostofi, and M. C. Payne, *J. Chem. Phys.* **122**, 084119 (2005).
- ²⁴⁶A. Nakata, D. R. Bowler, and T. Miyazaki, *Phys. Chem. Chem. Phys.* **17**, 31427 (2015).
- ²⁴⁷S. K. Burger and W. Yang, *J. Phys.: Condens. Matter* **20**, 294209 (2008).
- ²⁴⁸R. Z. Khaliullin and T. D. Kühne, *Phys. Chem. Chem. Phys.* **15**, 15746 (2013).
- ²⁴⁹R. Z. Khaliullin, J. VandeVondele, and J. Hutter, *J. Chem. Theory Comput.* **9**, 4421 (2013).
- ²⁵⁰H. Scheiber, Y. Shi, and R. Z. Khaliullin, *J. Chem. Phys.* **148**, 231103 (2018).
- ²⁵¹R. Z. Khaliullin, M. Head-Gordon, and A. T. Bell, *J. Chem. Phys.* **124**, 204105 (2006).
- ²⁵²R. Z. Khaliullin, E. A. Cobar, R. C. Lochan, A. T. Bell, and M. Head-Gordon, *J. Phys. Chem. A* **111**, 8753 (2007).
- ²⁵³R. Z. Khaliullin, A. T. Bell, and M. Head-Gordon, *J. Chem. Phys.* **128**, 184112 (2008).
- ²⁵⁴T. D. Kühne and R. Z. Khaliullin, *Nat. Commun.* **4**, 1450 (2013).
- ²⁵⁵P. R. Horn, E. J. Sundstrom, T. A. Baker, and M. Head-Gordon, *J. Chem. Phys.* **138**, 134119 (2013).
- ²⁵⁶H. Elgabarty and T. D. Kühne, "Tumbling with a limp: local asymmetry in water's hydrogen bond network and its consequences," *Phys. Chem. Chem. Phys.* (published online, 2020).
- ²⁵⁷F. Mauri, G. Galli, and R. Car, *Phys. Rev. B* **47**, 9973 (1993).
- ²⁵⁸S. Goedecker, *Rev. Mod. Phys.* **71**, 1085 (1999).
- ²⁵⁹J.-L. Fattebert and F. Gygi, *Comput. Phys. Commun.* **162**, 24 (2004).
- ²⁶⁰E. Tsuchida, *J. Phys.: Condens. Matter* **20**, 294212 (2008).
- ²⁶¹L. Peng, F. L. Gu, and W. Yang, *Phys. Chem. Chem. Phys.* **15**, 15518 (2013).
- ²⁶²E. Tsuchida, *J. Phys. Soc. Jpn.* **76**, 034708 (2007).
- ²⁶³J. Harris, *Phys. Rev. B* **31**, 1770 (1985).
- ²⁶⁴Y. Shi and R. Z. Khaliullin, "Robust linear-scaling optimization of compact localized orbitals in density functional theory," *arXiv:2004.05901* [cond-mat.mtrl-sci] (2020).
- ²⁶⁵T. D. Kühne, M. Krack, F. R. Mohamed, and M. Parrinello, *Phys. Rev. Lett.* **98**, 066401 (2007).
- ²⁶⁶T. D. Kühne and E. Prodan, *Ann. Phys.* **391**, 120 (2018).
- ²⁶⁷O. Schütt and J. VandeVondele, *J. Chem. Theory Comput.* **14**, 4168 (2018).
- ²⁶⁸G. Berghold, M. Parrinello, and J. Hutter, *J. Chem. Phys.* **116**, 1800 (2002).
- ²⁶⁹X.-P. Li, R. W. Nunes, and D. Vanderbilt, *Phys. Rev. B* **47**, 10891 (1993).
- ²⁷⁰H. Hellmann, *Einführung in die Quantenchemie* (F. Deuticke, 1937).
- ²⁷¹R. P. Feynman, *Phys. Rev.* **56**, 340 (1939).
- ²⁷²J. Almlöf and T. Helgaker, *Chem. Phys. Lett.* **83**, 125 (1981).
- ²⁷³M. Scheffler, J. P. Vigneron, and G. B. Bachelet, *Phys. Rev. B* **31**, 6541 (1985).
- ²⁷⁴P. Bendt and A. Zunger, *Phys. Rev. Lett.* **50**, 1684 (1983).
- ²⁷⁵M. C. Payne, M. P. Teter, D. C. Allan, T. A. Arias, and J. D. Joannopoulos, *Rev. Mod. Phys.* **64**, 1045 (1992).
- ²⁷⁶R. Car and M. Parrinello, *Phys. Rev. Lett.* **55**, 2471 (1985).
- ²⁷⁷G. Pastore, E. Smargiassi, and F. Buda, *Phys. Rev. A* **44**, 6334 (1991).
- ²⁷⁸F. A. Bornemann and C. Schütte, *Numer. Math.* **78**, 359 (1998).
- ²⁷⁹M. F. Camellone, T. D. Kühne, and D. Passerone, *Phys. Rev. B* **80**, 033203 (2009).
- ²⁸⁰C. S. Cucinotta, G. Miceli, P. Raiteri, M. Krack, T. D. Kühne, M. Bernasconi, and M. Parrinello, *Phys. Rev. Lett.* **103**, 125901 (2009).
- ²⁸¹D. Richters and T. D. Kühne, *JETP Lett.* **97**, 184 (2013).
- ²⁸²K. A. F. Röhrig and T. D. Kühne, *Phys. Rev. E* **87**, 045301 (2013).
- ²⁸³T. Spura, H. Elgabarty, and T. D. Kühne, *Phys. Chem. Chem. Phys.* **17**, 14355 (2015).
- ²⁸⁴P. Partovi-Azar, M. Berg, S. Sanna, and T. D. Kühne, *Int. J. Quantum Chem.* **116**, 1160 (2016).
- ²⁸⁵D. Richters and T. D. Kühne, *Mol. Simul.* **44**, 1380 (2018).
- ²⁸⁶R. Walczak, A. Savateev, J. Heske, N. V. Tarakina, S. Sahoo, J. D. Epping, T. D. Kühne, B. Kurpil, M. Antonietti, and M. Oschatz, *Sustainable Energy Fuels* **3**, 2819 (2019).
- ²⁸⁷S. Oschatz, J. Heske, S. Azadi, Z. Zhang, N. Tarakina, M. Oschatz, R. Khaliullin, M. Antonietti, and T. D. Kühne, *Sci. Rep.* **10**, 5832 (2020).
- ²⁸⁸S. Caravati, M. Bernasconi, T. D. Kühne, M. Krack, and M. Parrinello, *Appl. Phys. Lett.* **91**, 171906 (2007).
- ²⁸⁹S. Caravati, M. Bernasconi, T. D. Kühne, M. Krack, and M. Parrinello, *J. Phys.: Condens. Matter* **21**, 255501 (2009).
- ²⁹⁰S. Caravati, M. Bernasconi, T. D. Kühne, M. Krack, and M. Parrinello, *Phys. Rev. Lett.* **102**, 205502 (2009).
- ²⁹¹S. Caravati, D. Colleoni, R. Mazzarello, T. D. Kühne, M. Krack, M. Bernasconi, and M. Parrinello, *J. Phys.: Condens. Matter* **23**, 265801 (2011).
- ²⁹²J. H. Los, T. D. Kühne, S. Gabardi, and M. Bernasconi, *Phys. Rev. B* **87**, 184201 (2013).

- ²⁹³J. H. Los, T. D. Kühne, S. Gabardi, and M. Bernasconi, *Phys. Rev. B* **88**, 174203 (2013).
- ²⁹⁴S. Gabardi, S. Caravati, J. H. Los, T. D. Kühne, and M. Bernasconi, *J. Chem. Phys.* **144**, 204508 (2016).
- ²⁹⁵T. D. Kühne, M. Krack, and M. Parrinello, *J. Chem. Theory Comput.* **5**, 235 (2009).
- ²⁹⁶T. A. Pascal, D. Schärf, Y. Jung, and T. D. Kühne, *J. Chem. Phys.* **137**, 244507 (2012).
- ²⁹⁷C. Zhang, R. Z. Khaliullin, D. Bovi, L. Guidoni, and T. D. Kühne, *J. Phys. Chem. Lett.* **4**, 3245 (2013).
- ²⁹⁸C. Zhang, L. Guidoni, and T. D. Kühne, *J. Mol. Liq.* **205**, 42 (2015).
- ²⁹⁹C. John, T. Spura, S. Habershon, and T. D. Kühne, *Phys. Rev. E* **93**, 043305 (2016).
- ³⁰⁰D. Ojha, K. Karhan, and T. D. Kühne, *Sci. Rep.* **8**, 16888 (2018).
- ³⁰¹H. Elgabarty, N. K. Kaliannan, and T. D. Kühne, *Sci. Rep.* **9**, 10002 (2019).
- ³⁰²T. Clark, J. Heske, and T. D. Kühne, *Chem. Phys. Chem.* **20**, 2461 (2019).
- ³⁰³D. Ojha, N. K. Kaliannan, and T. D. Kühne, *Commun. Chem.* **2**, 116 (2019).
- ³⁰⁴H. Elgabarty, T. Kampfrath, D. J. Bonthuis, V. Balos, N. K. Kaliannan, P. Loche, R. R. Netz, M. Wolf, T. D. Kühne, and M. Sajadi, *Sci. Adv.* **6**, eaay7074 (2020).
- ³⁰⁵T. D. Kühne, T. A. Pascal, E. Kaxiras, and Y. Jung, *J. Phys. Chem. Lett.* **2**, 105 (2011).
- ³⁰⁶G. A. Ludueña, T. D. Kühne, and D. Sebastiani, *Chem. Mater.* **23**, 1424 (2011).
- ³⁰⁷B. Torun, C. Kunze, C. Zhang, T. D. Kühne, and G. Grundmeier, *Phys. Chem. Chem. Phys.* **16**, 7377 (2014).
- ³⁰⁸J. Kessler, H. Elgabarty, T. Spura, K. Karhan, P. Partovi-Azar, A. A. Hassanali, and T. D. Kühne, *J. Phys. Chem. B* **119**, 10079 (2015).
- ³⁰⁹P. Partovi-Azar and T. D. Kühne, *Phys. Status Solidi B* **253**, 308 (2016).
- ³¹⁰Y. Nagata, T. Ohto, M. Bonn, and T. D. Kühne, *J. Chem. Phys.* **144**, 204705 (2016).
- ³¹¹J. Lan, J. Hutter, and M. Iannuzzi, *J. Phys. Chem. C* **122**, 24068 (2018).
- ³¹²J. Kolafa, *J. Comput. Chem.* **25**, 335 (2004).
- ³¹³J. Dai and J. Yuan, *Eur. Phys. Lett.* **88**, 20001 (2009).
- ³¹⁴Y. Luo, A. Zen, and S. Sorella, *J. Chem. Phys.* **141**, 194112 (2014).
- ³¹⁵T. Musso, S. Caravati, J. Hutter, and M. Iannuzzi, *Eur. Phys. J. B* **91**, 148 (2018).
- ³¹⁶A. Jones and B. Leimkuhler, *J. Chem. Phys.* **135**, 084125 (2011).
- ³¹⁷L. Mones, A. Jones, A. W. Götz, T. Laino, R. C. Walker, B. Leimkuhler, G. Csányi, and N. Bernstein, *J. Comput. Chem.* **36**, 633 (2015).
- ³¹⁸B. Leimkuhler, M. Sachs, and G. Stoltz, "Hypoocoercivity properties of adaptive Langevin dynamics," *arXiv:1908.09363* (2019).
- ³¹⁹S. Nosé, *J. Chem. Phys.* **81**, 511 (1984).
- ³²⁰W. G. Hoover, *Phys. Rev. A* **31**, 1695 (1985).
- ³²¹J. Kussmann, M. Beer, and C. Ochsenfeld, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **3**, 614 (2013).
- ³²²D. M. Benoit, D. Sebastiani, and M. Parrinello, *Phys. Rev. Lett.* **87**, 226401 (2001).
- ³²³M. Tuckerman, B. J. Berne, and G. J. Martyna, *J. Chem. Phys.* **97**, 1990 (1992).
- ³²⁴G. J. Martyna, M. E. Tuckerman, D. J. Tobias, and M. L. Klein, *Mol. Phys.* **87**, 1117 (1996).
- ³²⁵P. F. Batcho and T. Schlick, *J. Chem. Phys.* **115**, 4019 (2001).
- ³²⁶K. Kitaura and K. Morokuma, *Int. J. Quantum Chem.* **10**, 325 (1976).
- ³²⁷Y. Mo, J. Gao, and S. D. Peyerimhoff, *J. Chem. Phys.* **112**, 5530 (2000).
- ³²⁸E. Gianinetti, M. Raimondi, and E. Tornaghi, *Int. J. Quantum Chem.* **60**, 157 (1996).
- ³²⁹T. Nagata, O. Takahashi, K. Saito, and S. Iwata, *J. Chem. Phys.* **115**, 3553 (2001).
- ³³⁰R. J. Azar, P. R. Horn, E. J. Sundstrom, and M. Head-Gordon, *J. Chem. Phys.* **138**, 084102 (2013).
- ³³¹R. Staub, M. Iannuzzi, R. Z. Khaliullin, and S. N. Steinmann, *J. Chem. Theory Comput.* **15**, 265 (2019).
- ³³²Y. Shi, H. Scheiber, and R. Z. Khaliullin, *J. Phys. Chem. A* **122**, 7482 (2018).
- ³³³Z. Luo, T. Frišić, and R. Z. Khaliullin, *Phys. Chem. Chem. Phys.* **20**, 898 (2018).
- ³³⁴T. D. Kühne and R. Z. Khaliullin, *J. Am. Chem. Soc.* **136**, 3395 (2014).
- ³³⁵Y. Yun, R. Z. Khaliullin, and Y. Jung, *J. Phys. Chem. Lett.* **10**, 2008 (2019).
- ³³⁶T. Fransson, Y. Harada, N. Kosugi, N. A. Besley, B. Winter, J. J. Rehr, L. G. M. Pettersson, and A. Nilsson, *Chem. Rev.* **116**, 7551 (2016).
- ³³⁷E. Ramos-Cordoba, D. S. Lambrecht, and M. Head-Gordon, *Faraday Discuss.* **150**, 345 (2011).
- ³³⁸A. Lenz and L. Ojamäe, *J. Phys. Chem. A* **110**, 13388 (2006).
- ³³⁹M. Reiher and J. Neugebauer, *J. Chem. Phys.* **118**, 1634 (2003).
- ³⁴⁰M. Reiher and J. Neugebauer, *Phys. Chem. Chem. Phys.* **6**, 4621 (2004).
- ³⁴¹F. Schiffmann, J. VandeVondele, J. Hutter, R. Wirz, A. Urakawa, and A. Baiker, *J. Phys. Chem. C* **114**, 8398 (2010).
- ³⁴²G. Schusteritsch, T. D. Kühne, Z. X. Guo, and E. Kaxiras, *Mol. Phys.* **96**, 2868 (2016).
- ³⁴³P. Sherwood, in *Modern Methods and Algorithms of Quantum Chemistry*, edited by J. Grotendorst (John von Neumann Institute for Computing, Forschungszentrum Jülich, Jülich, 2000), pp. 257–277.
- ³⁴⁴M. J. Field, P. A. Bash, and M. Karplus, *J. Comput. Chem.* **11**, 700 (1990).
- ³⁴⁵U. C. Singh and P. A. Kollman, *J. Comput. Chem.* **7**, 718 (1986).
- ³⁴⁶B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, *J. Comput. Chem.* **4**, 187 (1983).
- ³⁴⁷D. A. Case, T. E. Cheatham, T. Darden, H. Gohlke, R. Luo, K. M. Merz, A. Onufriev, C. Simmerling, B. Wang, and R. J. Woods, *J. Comput. Chem.* **26**, 1668 (2005).
- ³⁴⁸T. Laino, F. Mohamed, A. Laio, and M. Parrinello, *J. Chem. Theory Comput.* **1**, 1176 (2005).
- ³⁴⁹P. P. Ewald, *Ann. Phys.* **369**, 253 (1921).
- ³⁵⁰P. E. Blöchl, *J. Chem. Phys.* **103**, 7422 (1995).
- ³⁵¹T. Laino, F. Mohamed, A. Laio, and M. Parrinello, *J. Chem. Theory Comput.* **2**, 1370 (2005).
- ³⁵²D. Golze, M. Iannuzzi, M.-T. Nguyen, D. Passerone, and J. Hutter, *J. Chem. Theory Comput.* **9**, 5086 (2013).
- ³⁵³J. I. Siepmann and M. Sprik, *J. Chem. Phys.* **102**, 511 (1995).
- ³⁵⁴S. N. Steinmann, R. Ferreira De Moraes, A. W. Götz, P. Fleurat-Lessard, M. Iannuzzi, P. Sautet, and C. Michel, *J. Chem. Theory Comput.* **14**, 3238 (2018).
- ³⁵⁵L. K. Scarbath-Evers, R. Hammer, D. Golze, M. Brehm, D. Sebastiani, and W. Widdra, *Nanoscale* **12**, 3834 (2020).
- ³⁵⁶M. Dou, F. C. Maier, and M. Fyta, *Nanoscale* **11**, 14216 (2019).
- ³⁵⁷F. C. Maier, C. S. Sarap, M. Dou, G. Sivaraman, and M. Fyta, *Eur. Phys. J.: Spec. Top.* **227**, 1681 (2019).
- ³⁵⁸<https://www.cp2k.org/howto:ic-qmmm>
- ³⁵⁹C. I. Bayly, P. Cieplak, W. Cornell, and P. A. Kollman, *J. Phys. Chem.* **97**, 10269 (1993).
- ³⁶⁰D. Golze, J. Hutter, and M. Iannuzzi, *Phys. Chem. Chem. Phys.* **17**, 14307 (2015).
- ³⁶¹C. Campañá, B. Mussard, and T. K. Woo, *J. Chem. Theory Comput.* **5**, 2866 (2009).
- ³⁶²V. V. Korolkov, M. Baldoni, K. Watanabe, T. Taniguchi, E. Besley, and P. H. Beton, *Nat. Chem.* **9**, 1191 (2017).
- ³⁶³M. Witman, S. Ling, S. Anderson, L. Tong, K. C. Stylianou, B. Slater, B. Smit, and M. Haranczyk, *Chem. Sci.* **7**, 6263 (2016).
- ³⁶⁴M. Witman, S. Ling, S. Jawahery, P. G. Boyd, M. Haranczyk, B. Slater, and B. Smit, *J. Am. Chem. Soc.* **139**, 5547 (2017).
- ³⁶⁵M. Witman, S. Ling, A. Gladysiak, K. C. Stylianou, B. Smit, B. Slater, and M. Haranczyk, *J. Phys. Chem. C* **121**, 1171 (2017).
- ³⁶⁶L. Ciammaruchi, L. Bellucci, G. C. Castillo, G. M. D. Sánchez, Q. Liu, V. Tozzini, and J. Martorell, *Carbon* **153**, 234 (2019).
- ³⁶⁷<https://www.cp2k.org/howto:resp>
- ³⁶⁸T. A. Wesolowski, S. Shedge, and X. Zhou, *Chem. Rev.* **115**, 5891 (2015).
- ³⁶⁹C. Huang, M. Pavone, and E. A. Carter, *J. Chem. Phys.* **134**, 154110 (2011).
- ³⁷⁰Q. Wu and W. Yang, *J. Chem. Phys.* **118**, 2498 (2003).
- ³⁷¹L. Onsager, *J. Am. Chem. Soc.* **58**, 1486 (1936).
- ³⁷²J. A. Barker and R. O. Watts, *Mol. Phys.* **26**, 789 (1973).
- ³⁷³R. O. Watts, *Mol. Phys.* **28**, 1069 (1974).
- ³⁷⁴A. Klamt and G. Schuurmann, *J. Chem. Soc., Perkin Trans. 2* **1993**, 799.

- ³⁷⁵ J. Tomasi and M. Persico, *Chem. Rev.* **94**, 2027 (1994).
- ³⁷⁶ J. Tomasi, B. Mennucci, and R. Cammi, *Chem. Rev.* **105**, 2999 (2005).
- ³⁷⁷ M. Connolly, *Science* **221**, 709 (1983).
- ³⁷⁸ J.-L. Fattebert and F. Gygi, *J. Comput. Chem.* **23**, 662 (2002).
- ³⁷⁹ J.-L. Fattebert and F. Gygi, *Int. J. Quantum Chem.* **93**, 139 (2003).
- ³⁸⁰ O. Andreussi, I. Dabo, and N. Marzari, *J. Chem. Phys.* **136**, 064102 (2012).
- ³⁸¹ W.-J. Yin, M. Krack, B. Wen, S.-Y. Ma, and L.-M. Liu, *J. Phys. Chem. Lett.* **6**, 2538 (2015).
- ³⁸² W.-J. Yin, M. Krack, X. Li, L.-Z. Chen, and L.-M. Liu, *Prog. Nat. Sci.: Mater. Int.* **27**, 283 (2017).
- ³⁸³ D. A. Scherlis, J.-L. Fattebert, F. Gygi, M. Cococcioni, and N. Marzari, *J. Chem. Phys.* **124**, 074103 (2006).
- ³⁸⁴ M. Cococcioni, F. Mauri, G. Ceder, and N. Marzari, *Phys. Rev. Lett.* **94**, 145501 (2005).
- ³⁸⁵ T. Darden, D. York, and L. Pedersen, *J. Chem. Phys.* **98**, 10089 (1993).
- ³⁸⁶ U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee, and L. G. Pedersen, *J. Chem. Phys.* **103**, 8577 (1995).
- ³⁸⁷ A. Y. Toukmaji and J. A. Board, *Comput. Phys. Commun.* **95**, 73 (1996).
- ³⁸⁸ T. Laino and J. Hutter, *J. Chem. Phys.* **129**, 074102 (2008).
- ³⁸⁹ N. Yazdani, S. Andermatt, M. Yarema, V. Farto, M. H. Bani-Hashemian, S. Volk, W. Lin, O. Yarema, M. Luisier, and V. Wood, "Charge transport in semiconductors assembled from nanocrystals," [arXiv:1909.09739](https://arxiv.org/abs/1909.09739) [cond-mat.mtrl-sci] (2019).
- ³⁹⁰ M. H. Bani-Hashemian, S. Brück, M. Luisier, and J. VandeVondele, *J. Chem. Phys.* **144**, 044113 (2016).
- ³⁹¹ C. S. Praveen, A. Comas-Vives, C. Copéret, and J. VandeVondele, *Organometallics* **36**, 4908 (2017).
- ³⁹² S. Andermatt, M. H. Bani-Hashemian, F. Ducry, S. Brück, S. Clima, G. Pourtois, J. VandeVondele, and M. Luisier, *J. Chem. Phys.* **149**, 124701 (2018).
- ³⁹³ O. Schütt, P. Messmer, J. Hutter, and J. VandeVondele, *Electronic Structure Calculations on Graphics Processing Units* (John Wiley & Sons, 2016), pp. 173–190.
- ³⁹⁴ A. Lazzaro, J. VandeVondele, J. Hutter, and O. Schütt, in *Proceedings of the Platform for Advanced Scientific Computing Conference, PASC'17* (ACM, New York, NY, USA, 2017), pp. 3–1–3–9.
- ³⁹⁵ L. E. Cannon, "A cellular computer to implement the Kalman filter algorithm," Ph.D. thesis, Montana State University, 1969.
- ³⁹⁶ I. Sivkov, P. Seewald, A. Lazzaro, and J. Hutter, in *Proceedings of the International Conference on Parallel Computing, ParCo 2019*, Prague, Czech Republic, 10–13 September 2019.
- ³⁹⁷ A. Heinecke, G. Henry, M. Hutchinson, and H. Pabst, in *SC16: International Conference for High Performance Computing, Networking, Storage and Analysis, SC'16* (IEEE, Piscataway, NJ, USA, 2016).
- ³⁹⁸ I. Bethune, A. Glöss, J. Hutter, A. Lazzaro, H. Pabst, and F. Reid, "Advances in parallel computing," in *Proceedings of the International Conference on Parallel Computing, ParCo 2017, 12–15 September 2017* (SCAI, Bologna, Italy, 2017), pp. 47–56.
- ³⁹⁹ J. H. Los and T. D. Kühne, *Phys. Rev. B* **87**, 214202 (2013).
- ⁴⁰⁰ J. H. Los, S. Gabardi, M. Bernasconi, and T. D. Kühne, *Comput. Mat. Sci.* **117**, 7 (2016).
- ⁴⁰¹ V. Kapil, M. Rossi, O. Marsalek, R. Petraglia, Y. Litman, T. Spura, B. Cheng, A. Cuzzocrea, R. H. Meißner, D. M. Wilkins, B. A. Helfrecht, P. Juda, S. P. Bienvenue, W. Fang, J. Kessler, I. Poltavsky, S. Vandenbrande, J. Wieme, C. Corminboeuf, T. D. Kühne, D. E. Manolopoulos, T. E. Markland, J. O. Richardson, A. Tkatchenko, G. A. Tribello, V. Van Speybroeck, and M. Ceriotti, *Comput. Phys. Commun.* **236**, 214 (2019).
- ⁴⁰² M. Bonomi, D. Branduardi, G. Bussi, C. Camilloni, D. Provasi, P. Raiteri, D. Donadio, F. Marinelli, F. Pietrucci, R. A. Broglia, and M. Parrinello, *Comput. Phys. Commun.* **180**, 1961 (2009).
- ⁴⁰³ E. Riccardi, A. Lervik, S. Roet, O. Aarøen, and T. S. Erp, *J. Comput. Chem.* **41**, 370 (2019).
- ⁴⁰⁴ A. Ghysels, T. Verstraelen, K. Hemelsoet, M. Waroquier, and V. Van Speybroeck, *J. Chem. Inf. Model.* **50**, 1736 (2010).
- ⁴⁰⁵ T. Verstraelen, M. Van Houteghem, V. Van Speybroeck, and M. Waroquier, *J. Chem. Inf. Model.* **48**, 2414 (2008).
- ⁴⁰⁶ M. Brehm and B. Kirchner, *J. Chem. Inf. Model.* **51**, 2007 (2011).
- ⁴⁰⁷ L. P. Kadanoff and G. Baym, "Quantum statistical mechanics: Green's function methods in equilibrium and nonequilibrium problems," in *Frontiers in Physics* (W. A. Benjamin, 1962).
- ⁴⁰⁸ L. V. Keldysh, *Sov. Phys. JETP* **20**, 1018 (1964).
- ⁴⁰⁹ S. Datta, *Electronic Transport in Mesoscopic Systems*, Cambridge Studies in Semiconductor Physics and Microelectronic Engineering (Cambridge University Press, 1995), No. 3.
- ⁴¹⁰ J. Taylor, H. Guo, and J. Wang, *Phys. Rev. B* **63**, 245407 (2001).
- ⁴¹¹ M. Brandbyge, J.-L. Mozos, P. Ordejón, J. Taylor, and K. Stokbro, *Phys. Rev. B* **65**, 165401 (2002).
- ⁴¹² M. Luisier, A. Schenk, W. Fichtner, and G. Klimeck, *Phys. Rev. B* **74**, 205323 (2006).
- ⁴¹³ M. Luisier, *Chem. Soc. Rev.* **43**, 4357 (2014).
- ⁴¹⁴ M. H. Bani-Hashemian, "Large-scale nanoelectronic device simulation from first principles," Ph.D. thesis, ETH Zürich, 2016.
- ⁴¹⁵ C. S. Lent and D. J. Kirkner, *J. Appl. Phys.* **67**, 6353 (1990).
- ⁴¹⁶ S. Brück, M. Calderara, M. H. Bani-Hashemian, J. VandeVondele, and M. Luisier, in *International Workshop on Computational Electronics (IWCE)* (IEEE, 2014), pp. 1–3.
- ⁴¹⁷ M. Calderara, S. Brück, A. Pedersen, M. H. Bani-Hashemian, J. VandeVondele, and M. Luisier, in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC'15* (ACM, 2015), pp. 3–1–3–12.
- ⁴¹⁸ S. Brück, M. Calderara, M. H. Bani-Hashemian, J. VandeVondele, and M. Luisier, *J. Chem. Phys.* **147**, 074116 (2017).
- ⁴¹⁹ O. H. Nielsen and R. M. Martin, *Phys. Rev. Lett.* **50**, 697 (1983).
- ⁴²⁰ O. H. Nielsen and R. M. Martin, *Phys. Rev. B* **32**, 3780 (1985).
- ⁴²¹ A. Kozhevnikov, M. Taillefumier, S. Frisch, V. J. Pintarelli, Simon, and T. Schulthess, "Domain specific library for electronic structure calculations," <https://github.com/electronic-structure/SIRIUS>.
- ⁴²² A. D. Corso and A. M. Conte, *Phys. Rev. B* **71**, 115106 (2005).
- ⁴²³ O. K. Andersen, *Phys. Rev. B* **12**, 3060 (1975).
- ⁴²⁴ G. K. H. Madsen, P. Blaha, K. Schwarz, E. Sjöstedt, and L. Nordström, *Phys. Rev. B* **64**, 195134 (2001).
- ⁴²⁵ S. Lehtola, C. Steigemann, M. J. T. Oliveira, and M. A. L. Marques, *SoftwareX* **7**, 1 (2018).
- ⁴²⁶ V. I. Anisimov, J. Zaanen, and O. K. Andersen, *Phys. Rev. B* **44**, 943 (1991).
- ⁴²⁷ A. I. Liechtenstein, V. I. Anisimov, and J. Zaanen, *Phys. Rev. B* **52**, R5467 (1995).
- ⁴²⁸ P. Giannozzi, O. Andreussi, T. Brumme, O. Bunau, M. Buongiorno Nardelli, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, M. Cococcioni, N. Colonna, I. Carnimeo, A. Dal Corso, S. de Gironcoli, P. Delugas, R. A. DiStasio, A. Ferretti, A. Floris, G. Fratesi, G. Fugallo, R. Gebauer, U. Gerstmann, F. Giustino, T. Gorni, J. Jia, M. Kawamura, H.-Y. Ko, A. Kokalj, E. Küçükbenli, M. Lazzeri, M. Marsili, N. Marzari, F. Mauri, N. L. Nguyen, H.-V. Nguyen, A. Otero-de-la-Roza, L. Paulatto, S. Poncé, D. Rocca, R. Sabatini, B. Santra, M. Schlipf, A. P. Seitsonen, A. Smogunov, I. Timrov, T. Thonhauser, P. Umari, N. Vast, X. Wu, and S. Baroni, *J. Phys.: Condens. Matter* **29**, 465901 (2017).
- ⁴²⁹ P. Giannozzi, S. Baroni, N. Bonini, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, G. L. Chiarotti, M. Cococcioni, I. Dabo, A. Dal Corso, S. de Gironcoli, S. Fabris, G. Fratesi, R. Gebauer, U. Gerstmann, C. Gougousis, A. Kokalj, M. Lazzeri, L. Martin-Samos, N. Marzari, F. Mauri, R. Mazzarello, S. Paolini, A. Pasquarello, L. Paulatto, C. Sbraccia, S. Scandolo, G. Sclauzero, A. P. Seitsonen, A. Smogunov, P. Umari, and R. M. Wentzcovitch, *J. Phys.: Condens. Matter* **21**, 395502 (2009).
- ⁴³⁰ W. Gropp, T. Hoefler, R. Thakur, and E. Lusk, *Using Advanced MPI: Modern Features of the Message-Passing Interface* (MIT Press, Cambridge, MA, 2014).
- ⁴³¹ M. Frigo and S. G. Johnson, in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)* (IEEE, 1998), Vol. 3, pp. 1381–1384.
- ⁴³² <http://libint.valeev.net/>
- ⁴³³ A. Ramaswami, T. D. Kühne, and C. Plessl <https://github.com/pc2/fft3d-fpga>
- ⁴³⁴ C. Plessl, M. Platzner, and P. J. Schreier, *Inf. Spektrum* **38**, 396 (2015).

- ⁴³⁵P. Klavík, A. C. I. Malossi, C. Bekas, and A. Curioni, *Philos. Trans. R. Soc. London, Ser. A* **372**, 20130278 (2014).
- ⁴³⁶C. M. Angerer, R. Polig, D. Zegarac, H. Giefers, C. Hagleitner, C. Bekas, and A. Curioni, *Sustainable Comput.: Inf. Syst.* **12**, 72 (2016).
- ⁴³⁷A. Haidar, P. Wu, S. Tomov, and J. Dongarra, in *8th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems, ScalA17* (ACM, 2017).
- ⁴³⁸A. Haidar, S. Tomov, J. Dongarra, and N. J. Higham, in *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis, SC'18* (IEEE, 2018).
- ⁴³⁹K. Karhan, R. Z. Khaliullin, and T. D. Kühne, *J. Chem. Phys.* **141**, 22D528 (2014).
- ⁴⁴⁰V. Rengaraj, M. Lass, C. Plessl, and T. D. Kühne, "Accurate sampling with noisy forces from approximate computing," *Computation* **8**, 39 (2020).
- ⁴⁴¹K. Lejaeghere, V. Van Speybroeck, G. Van Oost, and S. Cottenier, *Crit. Rev. Solid State Mater. Sci.* **39**, 1 (2014).
- ⁴⁴²X. Gonze, F. Jollet, F. Abreu Araujo, D. Adams, B. Amadon, T. Applencourt, C. Audouze, J.-M. Beuken, J. Bieder, A. Bokhanchuk, E. Bousquet, F. Bruneval, D. Caliste, M. Côté, F. Dahm, F. Da Pieve, M. Delaveau, M. Di Gennaro, B. Dorado, C. Espejo, G. Geneste, L. Genovese, A. Gerossier, M. Giantomassi, Y. Gillet, D. R. Hamann, L. He, G. Jomard, J. Laflamme Janssen, S. Le Roux, A. Levitt, A. Lherbier, F. Liu, I. Lukačević, A. Martin, C. Martins, M. J. T. Oliveira, S. Poncé, Y. Pouillon, T. Rangel, G.-M. Rignanese, A. H. Romero, B. Rousseau, O. Rubel, A. A. Shukri, M. Stankovski, M. Torrent, M. J. Van Setten, B. Van Troeye, M. J. Verstraete, D. Waroquiers, J. Wiktor, B. Xu, A. Zhou, and J. W. Zwanziger, *Comput. Phys. Commun.* **205**, 106 (2016).
- ⁴⁴³K. Lejaeghere, G. Bihlmayer, T. Bjorkman, P. Blaha, S. Blugel, V. Blum, D. Caliste, I. E. Castelli, S. J. Clark, A. Dal Corso, S. de Gironcoli, T. Deutsch, J. K. Dewhurst, I. Di Marco, C. Draxl, M. Du ak, O. Eriksson, J. A. Flores-Livas, K. F. Garrity, L. Genovese, P. Giannozzi, M. Giantomassi, S. Goedecker, X. Gonze, O. Granas, E. K. U. Gross, A. Gulans, F. Gygi, D. R. Hamann, P. J. Hasnip, N. a. W. Holzwarth, D. Iu an, D. B. Jochym, F. Jollet, D. Jones, G. Kresse, K. Koepnik, E. Kucukbenli, Y. O. Kvashnin, I. L. M. Locht, S. Lubeck, M. Marsman, N. Marzari, U. Nitzsche, L. Nordstrom, T. Ozaki, L. Paulatto, C. J. Pickard, W. Poelmans, M. I. J. Probert, K. Refson, M. Richter, G.-M. Rignanese, S. Saha, M. Scheffler, M. Schlipf, K. Schwarz, S. Sharma, F. Tavazza, P. Thunstrom, A. Tkatchenko, M. Torrent, D. Vanderbilt, M. J. van Setten, V. Van Speybroeck, J. M. Wills, J. R. Yates, G.-X. Zhang, and S. Cottenier, *Science* **351**, aad3000 (2016).
- ⁴⁴⁴See <https://molmod.ugent.be/deltacodesdft> for deltacodesdft
- ⁴⁴⁵A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dulak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. B. Jensen, J. Kermode, J. R. Kitchin, E. L. Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. B. Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiotz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, and K. W. Jacobsen, *J. Phys.: Condens. Matter* **29**, 273002 (2017).
- ⁴⁴⁶G. Pizzi, A. Cepellotti, R. Sabatini, N. Marzari, and B. Kozinsky, *Comput. Mat. Sci.* **111**, 218 (2016).
- ⁴⁴⁷S. Curtarolo, G. L. W. Hart, M. B. Nardelli, N. Mingo, S. Sanvito, and O. Levy, *Nature Mater.* **12**, 191 (2013).
- ⁴⁴⁸<https://github.com/cp2k/cp2k-ci>
- ⁴⁴⁹<https://dashboard.cp2k.org/>
- ⁴⁵⁰<https://www.cp2k.org/static/coverage/>
- ⁴⁵¹Pseewald/Fpretify, 2019.
- ⁴⁵²G. Pizzi, V. Vitale, R. Arita, S. Bluegel, F. Freimuth, G. Géranton, M. Gibertini, D. Gresch, C. Johnson, T. Koretsune, J. Ibanez, H. Lee, J.-M. Lihm, D. Marchand, A. Marrazzo, Y. Mokrousov, J. I. Mustafa, Y. Nohara, Y. Nomura, L. Paulatto, S. Ponce, T. Ponweiser, J. Qiao, F. Thöle, S. S. Tsirkin, M. Wierzbowska, N. Marzari, D. Vanderbilt, I. Souza, A. A. Mostofi, and J. R. Yates, *J. Phys.: Condens. Matter* **32**, 165902 (2019).
- ⁴⁵³<https://pre-commit.com/>
- ⁴⁵⁴<https://packages.debian.org/search?keywords=cp2k>
- ⁴⁵⁵<https://apps.fedoraproject.org/packages/cp2k>
- ⁴⁵⁶<https://aur.archlinux.org/packages/cp2k/>
- ⁴⁵⁷https://spack.readthedocs.io/en/latest/package_list.html#cp2k
- ⁴⁵⁸<https://github.com/easybuilders/easybuild-easyconfigs>